CDS

TECHNICAL MEMORANDUM NO. CIT-CDS 97-011
August, 1997

"Robotic Manipulation with Flexible Link Fingers"

Sudipto Sur

Control and Dynamical Systems
California Institute of Technology
Pasadena, California 91125

Robotic Manipulation with Flexible Link Fingers

Thesis by
Sudipto Sur

Technical Report CDS 97-011
for the
Engineering and Applied Sciences

California Institute of Technology
Pasadena, California

1997

(Thesis defended January 20, 1997)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A robot manipulator is a spatial mechanism consisting essentially of a series of bodies, called "links", connected to each other at "joints". The joints can be of various types: revolute, rotary, planar, prismatic, telescopic or combinations of these. A serial connection of the links results in an open-chain manipulator. Closed-chain manipulators result from non-serial (or parallel) connections between links. Actuators at the joints of the manipulator provide power for motion.

A robot is usually not designed for a very specific or repetitive task which can be done equally well by task-specific machines. Its strength lies in its ability to handle a range of tasks by virtue of being "re-programmable". Therefore, in addition to the mechanical hardware two other elements are integral to the description of a robot: sensors and control. With the advent of micro-electronics and digital computers the availability of sensors is ever increasing and the control is usually done by software executed by computers which also collect the sensory data. It is possible to model quite accurately, the dynamics of robot manipulators for purposes of control. However, for most practical robots the models are complex and numerically intensive to calculate in real-time.

Traditional analyses of robot manipulators consider the whole mechanism to be rigid. Relaxation of the assumption of rigidity leads to further complication of the dynamics of the manipulator, leading to more difficulties in control. The overall motion of the manipulator is augmented by additional motion due to the dynamics of flexibility which must be considered. Sensing is also made more difficult. However, the ability to control robots with significant structural flexibilities, referred to as flexible robots in the rest of this thesis, influences robotics in many ways. It allows for consideration of new applications, observance of less conservative structural design and performance enhancements in certain classes of robotic tasks, which will be addressed in greater detail in the sections which follow.

1.1 Motivation

Our original motivation for doing work on flexible manipulators comes from the field of medicine. Endoscopes, used for surgically non-invasive examination of the alimentary canal could be enormously enhanced by attaching robotic fingers at their

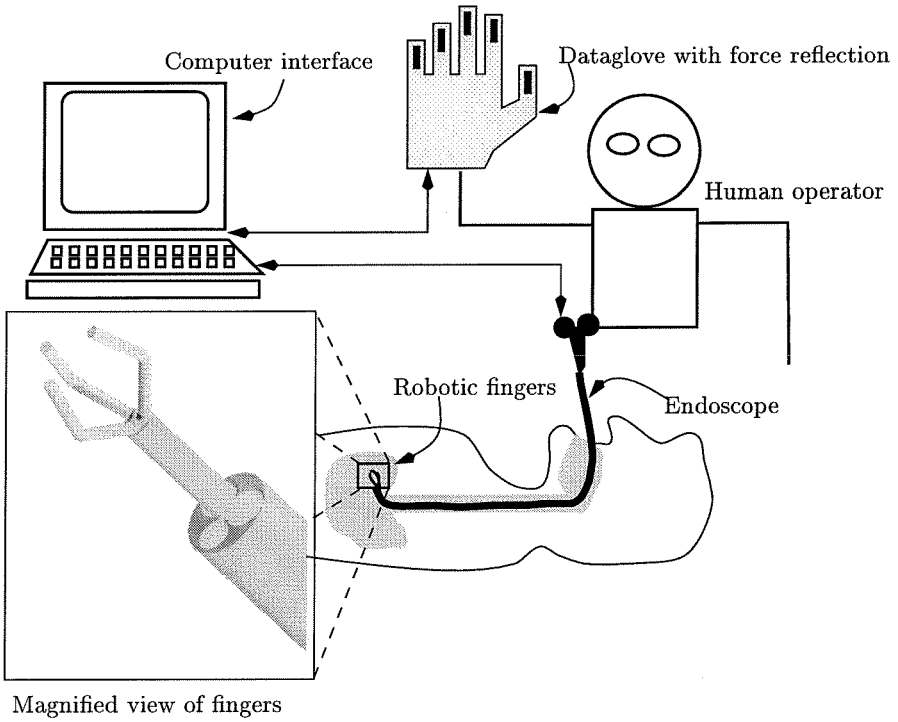


Figure 1.1 A possible "telesurgery" setup.

tip, controlled teleoperatically by a surgeon wearing a "dataglove" (see Figure 1.1). This would not only aid in examination but could conceivably be used for manipulation of bodies like tumors and polyps and even in the performance of surgical procedures. It is estimated that a lumen 3 mm in diameter would be available to accommodate these fingers at the tip of the endoscope into which the fingers would have to fit during insertion of the device to prevent snagging and interference. If we consider a set of three fingers, which would be the minimum required for sufficient dexterity, the dimensions of the fingers make it difficult to ensure sufficient rigidity. This is a scenario in which *flexibility is unavoidable.* Flexible manipulators can perform better than rigid manipulators in certain tasks where control of both positions and forces are desired. Being that the nature of the manipulation task for the endoscopic fingers requires simultaneous force and position control, it may indeed be that in this case, additionally, *flexibility is desirable.* This work addresses both these aspects of flexible robotics.

There are more commonplace scenarios than the fairly esoteric one mentioned above that encounter flexibility. With improvements in electric motor technology modern manipulators can not only carry or move bigger loads, but can do so with faster accelerations. This can cause flexure even in nominally rigid manipulators, thus creating performance shortcomings. To avoid dealing with flexibilities robots are usually over-designed. Thus present generation manipulators are limited to carrying loads no more that 5-10% of their weight. As an example the Cincinatti-

Milacron T3R3 robot weighs 1800 kg but can carry no more than 23 kg [9]. The ability to control flexibility immediately translates to a reduction in weight. Reduction in weight of manipulators is beneficial for the reasons mentioned below.

- Lower energy consumption: reduced inertias of the lighter robots require less power to produce the same accelerations and load-carrying capacity as heavier robots.
- Smaller actuators: reduced power requirements can be satisfied by smaller actuators, which are generally cheaper.
- Safer operation: collision of the smaller inertia causes lesser damage.
- Lower mounting strength: this is relevant to gantry and wall mounted robots.
- Simplification of drive mechanism: lighter links can be direct-driven, given the improving power to weight (or size) ratios for electric motors. This would eliminate the need for drive elements like gears, which introduce backlash.
- Faster operation: greater accelerations can be achieved for lighter robots. For certain modern applications of robots, for example, the testing of microchip and printed circuit contacts, a high speed of operation is very important because of the large number of operations needed to be carried out. As the task does not require a rigid robot (there being no loads to carry) the overhead incurred due to the inability to control flexibility is significant.
- Significant cost reduction in deployment of space robots: robots like the space shuttle arm and the robots envisaged for the construction and maintenance of the international space station have to be boosted into orbit. Considering that about 95% of the takeoff weight of the space shuttle is the weight of the fuel, it is evident that the savings in fuel due to any reduction in the weight of the payload are significant.

From the above discussion it is clear that there are a variety of applications which would benefit significantly from the ability to control robots which are light and fast, and therefore naturally flexible. There is also a class of applications where flexibility is not optional. One such is the endoscopic robot finger example mentioned previously. Micro-robots, which are robots crafted out of polysilicon wafers by techniques similar to the fabrication of integrated circuits [43, 44] are another example of robots which are necessarily flexible. The sizes of these robots is of the order of one cubic μm. Micro-robots contain micro-motors, micro-sensors and integrated circuits all in a work space which can be only made visible by microscopes. The forces due to surface-tension, pressure-impact as well as magneto- and electrostatic forces can be very significant at these scales, and can cause large flexure. The range of applications envisaged for micro-robots is vast. Use in medicine ranges from drug delivery [19] to delicate operations in neurosurgery and opthalmology which need to be done with extreme precision [12]. In bio-technology micro-robots will provide a potent tool to manipulate individual cells. In industry micro-robots could

be used for integrated circuit production, for finding errors on semiconductor dice, fabrication and maintenance of high precision tools and even for the fabrication of even smaller robots–the nanorobots.

Increasingly, the tasks performed by robots involve physical interaction with their environment. This naturally gives rise to interactive forces between the robot and its environment. The task of the controller increases from merely position control to simultaneous force and position control—called hybrid control. The mechanical compliance introduced by flexibility is useful in hybrid control in two respects. First, the flexible links can themselves be used for sensing the forces and torques. Second and more importantly, the compliance in the structure increases the robustness properties of the manipulator. This can be very significant because it is difficult, if not impossible to predict all events which might happen in the real working environment of the robot, and which will have an effect on the robot. Related to this is the issue of contact transition—the transition from a free state to a state where the robot is in physical contact with some component of its environment. This process often exhibits large jitter, which is difficult to control. Jitter causes wear and tear on the mechanism and its environment, and large force transients. Structural compliance in the manipulator is one method which can be used for attenuating jitter.

Tasks involving multi-robot cooperation is another area of interest in modern robotics to which many elements of the foregoing discussion are relevant. Control of multiple interacting robots typically requires tools from hybrid control. Robotic grasping, a special case of multi-robot cooperation, is a subject of ongoing research in the robotics community. In addition to its many practical applications, multi-fingered hands are an excellent application for developing new ideas in intelligent control of complex dynamical systems. Teleoperated robotic hands are an important example of man-machine systems which requires research in user-interfaces, hierarchical control and control of complex systems with a human in the loop.

The work described in this thesis is an effort to incorporate flexibility into the robot dynamics and control. It is motivated by the need to decrease the size and weight of robot manipulators, while at the same time increasing performance. Rather than try to design away flexibilities because of their complexity, it is important to gain an understanding of how to *use* flexibilities to increase the performance of a system. Examples of applications to which this work applies include space manipulation tasks, non-invasive surgical techniques and micro-robots.

1.2 Aspects of the Problem and Previous Work

Robotic manipulation with flexible fingers requires the blending together of concepts from a multitude of different disciplines. The theoretical analysis in this thesis draws on ideas from research on hybrid force/position control, grasping with multi-fingered hands, flexible structures, modeling and control of joint and link flexibility in robots and singular perturbation theory. Fabrication of apparatus for the experimental implementation required a multistage evolution of design ideas, consideration of sensing technology and issues in real-time computer control.

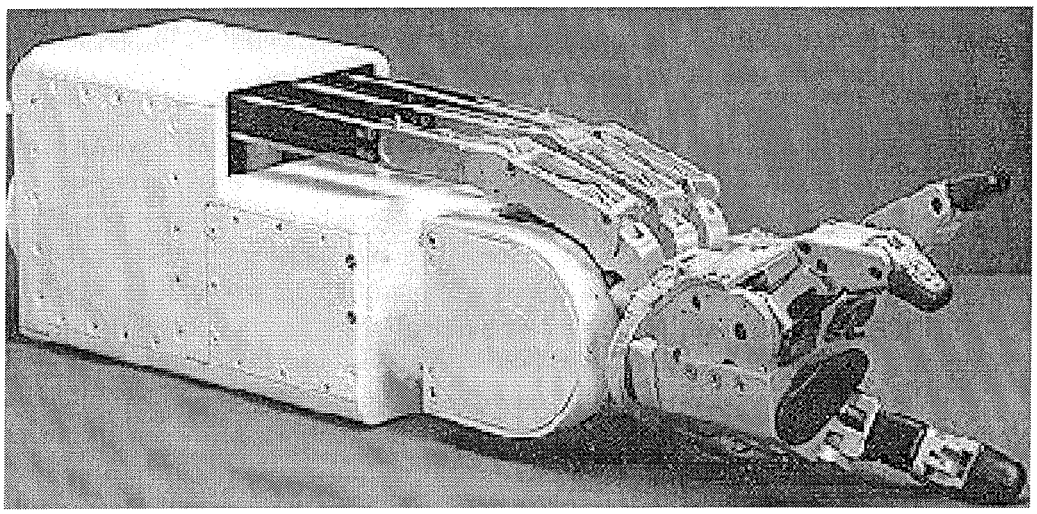


Figure 1.2 The Jameson Hand (JH-2). (Courtesy of NASA Johnson Space Center)

There are many robotic tasks which cannot be defined solely in terms of the motion of the end-effector (or tip), one such being tasks which are characterized by physical contact between the end-effector and a constraint surface. Combining force and motion (or position) control in an unknown environment was first proposed by Craig and Raibert [5, 45]. This was termed hybrid force/position control. Zhang and Paul [61] modified the control scheme from a Cartesian to a joint space formulation. In both cases it was always possible to independently analyze the force information and the position information and then combine them at the final stage when they had already been converted to joint torques. Since then there have been many modifications, enhancements and interpretations of the basic hybrid control scheme proposed by Craig and Raibert. Some of the better known variants are referred to as impedance control [13, 14, 15], compliance control [31, 35] and stiffness control [48]. Though the field has been actively researched for close to twenty years there is still disagreement among researchers about the proper formulation of the problem [8] and as yet there is no global formulation.

Multi-fingered robot hands have been an active research area for over ten years. Early designs included a three-fingered hand built by Okada which was capable of manipulating a bar using a pre-programmed sequence of motions [41]. The JPL/Stanford hand [57], the Utah/MIT hand [20] and the Jameson hand (refer to Figure 1.2) are more recent designs. Both these mechanisms were roughly anthropomorphic with tendon driven fingers. Control was implemented by individual joint servos which move the fingers to specified joint configurations.

Analysis of the kinematics, dynamics and control of multi-fingered hands is a mature field. Fundamental work in grasping was done by Salisbury [47] and Kerr [21]. Derivations of the dynamics of manipulation and formulation of controller were given by Li *et al.* [29, 30]. Most work in grasping consider very simple contact models;

extensions to finger rolling and compliant contacts can be found in [4, 37, 38]. All the work in grasping mentioned assumes that the object and the fingers are rigid and their geometry is completely known.

Control of robots with flexible links has concentrated primarily on position control of the end-effector of a flexible robot in a point-to-point positioning task. A single link flexible robot was investigated at the theoretical as well as experimental level by Cannon and Schmitz [2] in 1984. The control of multiple-link, flexible robots is considerably more difficult and is an area of active research (see [9] for a survey). Experimental work in this area is particularly difficult to find, in part due to some of the theoretical difficulties inherent in the problem.

Considerably less work is available on the dynamics and control of flexible link robots in contact with the environment. Some initial work has been performed by Latornell and Cherchas, who have studied force and motion control of a single flexible manipulator link [26]. In addition, Kozel, Koivo and Mahil have studied the force relationships between flexible manipulators in contact with their environment [25], Mills has studied the stability of a flexible link manipulator during constrained motion tasks using a singular perturbation approach [36], and Matsuno, Sakawa, and Asano have studied hybrid position/force control under quasi-static assumptions [32].

Considerable research effort has also been expended on the modeling of structural flexibility in a form suitable for application to flexible robot modeling and control. Most models of flexible links are finite dimensional models, either derived from truncating the number of modes of an infinite dimensional model [9], or by discretizing the links [59, 60]. Cetinkunt and Yu have addressed the issue of selecting the shape and number of mode functions in developing finite order models for control of a flexible robot arm [3].

Modeling however is only the first step in the study. Analysis of the model is as important and essential to the ultimate goal of controller design, and often more difficult than constructing the model itself. Mechanical systems with flexibilities have often been analysed using tools from singular perturbation theory. The perturbation parameters (ϵ) in these analyses are typically the inverse of the stiffness of the flexible mechanism or the inverse of the stiffness weighted by a factor depending on the mass (see for example [22]). Singular perturbation techniques have been used previously to deal with joint flexibility in robotic manipulators [52]. In the literature there is also discussion of perturbation techniques for flexible link manipulators [9]. Again, the perturbation parameters used in these analyses has always been scaled from the inverse of the stiffness associated with the manipulator. Thus the reduced system ($\epsilon = 0$) is rigid. Small values of the perturbation parameter correspond to systems which have a "small" amount of flexibility. Thus these analyses present results useful for systems which have a small amount of flexibility. The analysis of systems exhibiting significant flexibilities has not so far been undertaken.

Singular perturbation theory has been used in the past for application to control problems. The most notable contribution of singular perturbation theory to control efforts has been the simplification of dynamic models. It has been used to rigorously justify the neglect of small time constants, masses, capacitances and other parasitic

parameters. Furthermore, the separation into a reduced, slow, outer system and a boundary layer, fast inner system which is a contribution of singular perturbation analysis can be used to design stages of a control system depending on the performance desired. Kokotovic discusses typical applications of singular perturbation techniques to control problems and provides a review of the technique in [24].

1.3 Contributions of this Work

Introducing flexibility in robots introduces two main problems:

- The kinematics of the robot become a function of the state of the forces acting on the robot, including most importantly, those being applied by the robot itself. Deformations in the robots mechanical structure caused by the forces have to be considered.
- Controllers must be extended to not only achieve the primary manipulation task, but also to stabilize flexible modes in the structure of the system. The nonlinear nature of the problem requires use of techniques in nonlinear control theory to provide a useful analysis of the system.

Given the above points this thesis answers the following questions:

1. Are there provably stable control laws which can be used to control flexible link robots with significant flexibility (beyond the linear range) in tasks requiring physical interaction with the environment?
2. Can these laws be implemented on real systems with current technology?
3. Are there guidelines for design which make it easier to control flexible link robots?
4. Given that flexible robots are more difficult to control than rigid robots is there a case for using flexible robots?
5. What are the tradeoffs in using flexible robots instead of rigid robots?
6. How much flexibility is good?

The first three questions are answered by the theory developed during the course of the research on which this thesis is based. We develop control laws which are provably stable, and which can be used to control flexible link robots in hybrid force/position control tasks. The tool of analysis is singular perturbation theory and we are able to formulate the flexibility in a way which allows us to treat significant flexibility. These laws can be implemented on real systems with current technology and are indeed implemented in the experimental part of our research. The process of modeling and proving stability also provides some pointers which can be used to design flexible robots which are easier to control. This fits in with the general trend in design and development of new products where the control system must

be considered part of the design process rather than an accessory after the design is complete.

The last three questions are answered by the experimental part of this research. A planar, two fingered hand, with the last link flexible in each finger, was fabricated for the experimentation. The setup itself is novel in that it is reconfigurable and can be used as a test bed for experiments in robotics. Experiments in human-robot interaction and the effect of flexible tendon actuation have been conducted on the same setup. The setup was designed to be scalable in line with our original motivation of endoscopic robot fingers.

From experiments it is clear that in case of grasping there are advantages to be gained by using flexible manipulators. A framework for evaluation of performance is set up allowing us to quantify to some extent the performance gains in force regulation versus performance degradation in position tracking due to the use of flexible links. It is also clear from the experimental work that the controllers developed in theory are indeed applicable to the real situation and can be implemented with current technology.

We also present data from simulations to supplement the experimental data.

1.4 Organization of the Thesis

In the succeeding chapters we present first the basics of robot dynamics and control, the more complex analysis used for treating multi-robot cooperation and modeling of flexibilities in Chapter 2. Singular perturbation theory basics and its application by us to the modeling of flexible robots is presented in Chapter 3. In Chapter 4 we describe and prove the stability of the controllers developed for controlling flexible robots. Chapter 5 is a compilation of simulation results. In Chapter 6 we present experimental data and discuss the results. Chapter 7 is a summary of the work presented in this thesis and a collection of ideas for future work. Appendix A is a description of the reconfigurable, multi-robot testbed designed and fabricated during the course of this work, and used for the experimental work.

Chapter 2

Dynamics and Control of Cooperative Multirobot Systems in Contact Tasks

Dealing with multiple robots in cooperative tasks is more involved than being able to deal with multiple individual robots separately. Interactions between robots lead naturally to situations where we need to control both the relative positions of the robots and the forces of interaction between them. The ability to deal with workspace constraints and resolve kinematic redundancy is required to analyse and control such systems. In this chapter we describe the dynamic equations for multiple robots in contact tasks. We start with the dynamics and control of individual robots which form the simplest robot control problems. The much harder problem of modeling of constraints and simultaneous force-position hybrid control are discussed next. The kinematics and dynamics of multirobot systems follow. Finally, we set up the equations for robots with flexibilities.

2.1 Simple Robot Dynamics and Control

There are many different methods for deriving the dynamics of mechanical systems. The method we outline in this section stems from a Lagrangian analysis. The advantage of this approach is that Lagrangian theory deals with constrained systems, and is able to formulate the problem completely without knowledge of the precise form of the constraints. It also reduces the problem to the smallest possible set of equations, of motion and of constraint, by the introduction of generalized coordinates. As a result we get dynamic equations which are in the most convenient form for our further analysis. Lagrange's method is based on the energy properties of the system. The resulting equations can be computed in closed form, allowing detailed analysis of the system. A thorough discussion of the general principles of Lagrangian analysis in mechanics can be found in Rosenberg [46] or Goldstein [11]. Their use in robotics is described in Murray, Li and Sastry [40].

2.1.1 The Lagrangian approach for deriving robot dynamics

Lagrange's equations for mechanical systems are derived by considering the constraints on the system very explicitly. This has obvious advantages for deriving the

dynamics of manipulators, each link of the manipulator giving a set of constraints for the system. The following concepts are basic to the Lagrangian approach.

Generalized coordinates: A set of coordinates $\{q_1, q_2, \ldots, q_n\} \in \mathbb{R}^n$ are called the generalized coordinates for a system if all n of them are necessary and sufficient to define the configuration of the system uniquely. In other words this is a minimal set of coordinates for a system.

A system with N particles and no constraints has $3N$ independent coordinates (or degrees of freedom). The imposition of L holonomic constraints on this system reduces the number of required coordinates (or generalized coordinates) by L. (For the moment we will use this as a definition of holonomic constraints.) The $3N - L$ remaining coordinates contain the constraints implicitly in them. This is only true in the case of holonomic constraints, and imposition of nonholonomic constraints will not in general cause a reduction.

For a revolute jointed open-chain robot (see Figure 2.1), the set of joint angles (Θ say) forms a set of convenient generalized coordinates. Here we use the term joint angles in a broader sense to include also the displacements in Cartesian manipulators. In the case of manipulators with workspace constraints it is usually extremely difficult to find generalized coordinates. As will be shown later in this chapter we use a slightly modified approach in that case.

Virtual displacements and virtual work: Consider L holonomic constraints on a system of N particles:

$$f_r(u_1, u_2, \ldots, u_N, t) = 0, \qquad (r = 1, 2, \ldots, L),$$

where u_i is the position of the ith particle and t represents time. The differential form of this system is

$$\sum_{i=1}^{N} \frac{\partial f_r}{\partial u_i} du_i + \frac{\partial f}{\partial t} dt = 0, \qquad (r = 1, 2, \ldots, L),$$

which is a set of L first-order differential equations. We can write these as

$$\sum_{i=1}^{N} A_{ir} du_i + A_i dt = 0, \qquad (r = 1, 2, \ldots, L). \tag{2.1}$$

This is the *Pfaffian form* of the constraint equations. Note that this is the general form of equality constraints in classical mechanics. If they are integrable then the constraints are holonomic; else they are nonholonomic.

The set of infinitesimal quantities δu_i $(i = 1, \ldots, N)$ which satisfy the equations

$$\sum_{i=1}^{N} A_{is} \delta u_i = 0, \qquad (r = 1, 2, \ldots, L),$$

are called virtual displacements, and $\delta u = (\delta u_1, \ldots, \delta u_N)$ is called the virtual dis-

placement vector. Virtual displacements are consistent with the forces and the constraints imposed on the system *at a given instant t.*

The virtual work done by a force is the work it does in a virtual displacement.

D'Alembert's principle of virtual work: This principle states that the net virtual work done by the forces of constraint is zero.

We denote by $q = (q_1, q_2, \ldots, q_n)$ a set of generalized coordinates for the system. The Lagrangian for a mechanical system is defined to be

$$L(q, \dot{q}) = T(q, \dot{q}) - V(q),$$

where, T is the kinetic energy of the system and V the potential energy.

For constrained systems satisfying D'Alembert's principle we can write in generalized coordinates

$$\sum_{i=1}^{n} \left[\frac{d}{dt} \frac{\partial T}{\partial \dot{q}_i} - Q_i \right] \delta q_i = 0, \tag{2.2}$$

where the Q_i are generalized forces. Note that because q_i need not in general have units of length, Q_i also does not necessarily have units of force. However, the product $Q_i \dot{q}_i$ always has units of power. If we restrict the constraints to be holonomic, it is possible (though very difficult sometimes) to find sets of independent coordinates that contain the constraint conditions implicitly. The joint angles of open-chain manipulators are such coordinates. Considering q to be such a set any virtual displacement δq_j is independent of any other δq_k. Therefore equation (2.2) leads to

$$\frac{d}{dt} \frac{\partial T}{\partial \dot{q}_i} - \frac{\partial T}{\partial q_i} = Q_i. \tag{2.3}$$

We now split the generalized forces Q_i into two components: Q_{p_i} derived from the scalar potential energy field V and Q_{e_i} which make up the remaining magnitude of the force. Therefore

$$Q_{p_i} = -\nabla_i V,$$

and equation (2.3) becomes

$$\frac{d}{dt} \frac{\partial (T - V)}{\partial \dot{q}_i} - \frac{\partial (T - V)}{\partial q_i} = Q_{e_i}.$$

From our definition of the Lagrangian we can write this as

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = Q_{e_i}. \tag{2.4}$$

For the whole system we can write a vector equation

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = Q_e. \tag{2.5}$$

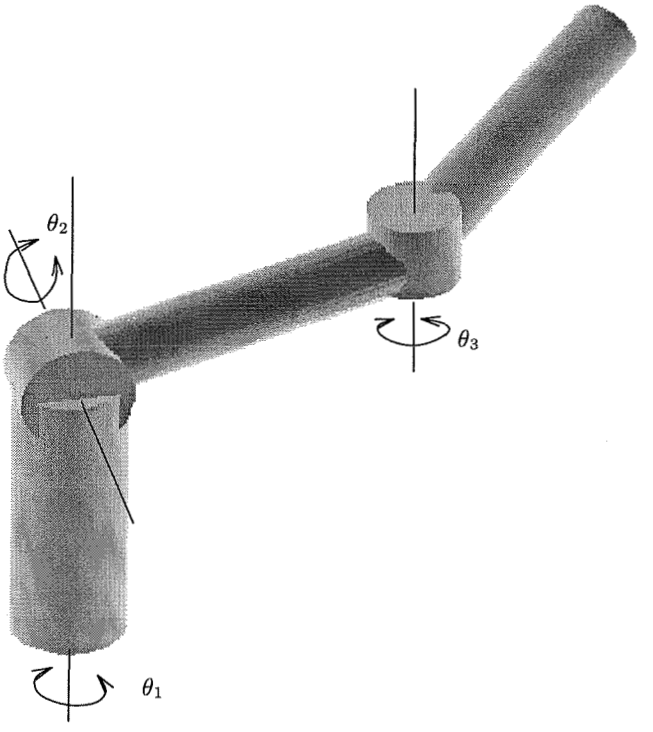


Figure 2.1 An open-chain manipulator.

In the sequel we shall call equations (2.4, 2.5) Lagrange's equation.

2.1.2 Dynamics of open-chain manipulators

We consider a n-link open-chain manipulator with joint angles $\Theta \in \mathbb{R}^n, \Theta = (\theta_1, \ldots, \theta_n)$. Denoting the manipulator *inertia matrix* by $M(\Theta)$, the kinetic energy of the manipulator can be written as

$$T(\Theta, \dot{\Theta}) = \frac{1}{2}\dot{\Theta}^T M(\Theta)\dot{\Theta}.$$

We use the above as a definition for the inertia matrix. We write the potential energy as $V(\Theta)$. If, for example, we considered only a gravitational field giving rise to the potential, then we could write

$$V(\Theta) = \sum_{i=1}^{n} m_i g h_i(\Theta),$$

where m_i is the mass of the ith link, h_i the height of its center of mass and g the gravitational constant. The Lagrangian can now be written

$$L(\Theta, \dot{\Theta}) = \frac{1}{2}\dot{\Theta}^T M(\Theta)\dot{\Theta} - V(\Theta).$$

It is more convenient to write the kinetic energy as a sum,

$$L(\Theta, \dot{\Theta}) = \frac{1}{2} \sum_{i,j=1}^{n} M_{ij}(\Theta)\dot{\theta}_i\dot{\theta}_j - V(\Theta). \tag{2.6}$$

To get the equations of motion we substitute the above into Lagranges' equations (2.4). We will use Q_{e_i} in that equation to represent actuator torques and other non-conservative, generalized forces acting on the ith joint. The terms in Lagranges' equations are (for joint i)

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_i} = \frac{d}{dt}\left(\sum_{j=1}^{n} M_{ij}\dot{\theta}_j\right) = \sum_{j=1}^{n}\left(M_{ij}\ddot{\theta}_j + \dot{M}_{ij}\dot{\theta}_j\right)$$

$$\frac{\partial L}{\partial \theta_i} = \frac{1}{2}\sum_{j,k=1}^{n} \frac{\partial M_{kj}}{\partial \theta_i}\dot{\theta}_k\dot{\theta}_j - \frac{\partial V}{\partial \theta_i}$$

M_{ij} is the (i,j)th element of the mass matrix. $\dot{M}_{ij}$ can be written as

$$\dot{M}_{ij} = \sum_{k=1}^{n} \frac{\partial M_{ij}}{\partial \theta_k}\dot{\theta}_k.$$

Using the above we get

$$\sum_{j=1}^{n} M_{ij}(\Theta)\ddot{\theta}_j + \sum_{j,k=1}^{n}\left(\frac{\partial M_{ij}}{\partial \theta_k}\dot{\theta}_k\dot{\theta}_j - \frac{1}{2}\frac{\partial M_{kj}}{\partial \theta_i}\dot{\theta}_k\dot{\theta}_j\right) + \frac{\partial V}{\partial \theta_i}(\Theta) = Q_{e_i},$$

which can be rearranged as

$$\sum_{j=1}^{n} M_{ij}(\Theta)\ddot{\theta}_j + \sum_{j,k}^{N} \Gamma_{ijk}\dot{\theta}_j\dot{\theta}_k + \frac{\partial V}{\partial \theta_i}(\Theta) = Q_{e_i}, \tag{2.7}$$

$$\Gamma_{ijk} = \frac{1}{2}\left(\frac{\partial M_{ij}(\Theta)}{\partial \theta_k} + \frac{\partial M_{ik}(\Theta)}{\partial \theta_j} - \frac{\partial M_{kj}(\Theta)}{\partial \theta_i}\right). \tag{2.8}$$

Equation (2.7) is one of the n second order differential equations required to describe the dynamics of the robot. The first term arises due to the acceleration of the joints and is called the inertial term of the dynamics. It accounts for the inertial forces in the robot.

The terms quadratic in the joint velocities are due to centrifugal and Coriolis forces. These forces exist because of non-inertial frames which arise naturally from the use of generalized coordinates. The cross-terms ($\dot{\theta}_i\dot{\theta}_j, i \neq j$) are the Coriolis terms and the others ($\dot{\theta}_i^2$) are the centrifugal terms. The functions Γ_{ijk} are called Christoffel symbols corresponding to the inertia matrix $M(\Theta)$.

The last term on the left hand side of equation (2.7) are the potential forces. Finally, Q_{e_i} represents the external forces on the joint.

To make explicit the actuator forces we represent the actuator generated generalized forces at the ith joint by τ_i. All other generalized forces acting on the ith joint, including conservative forces arising from a potential and frictional forces are represented by $-N_i(\Theta, \dot{\Theta})$. For example, for a manipulator with viscous friction in the ith joint

$$-N_i(\Theta, \dot{\Theta}) = -\frac{\partial V}{\partial \theta_i} - k_{fi}\dot{\theta}_i,$$

where k_{fi} is the damping coefficient.

To write the equations of motion for the whole manipulator in vector form we define the *Coriolis matrix*, $C(\Theta, \dot{\Theta}) \in \mathbb{R}^{n \times n}$ elements of which are given by

$$C_{ij}(\Theta, \dot{\Theta}) = \sum_{k=1}^{n} \Gamma_{ijk}\dot{\theta}_k. \tag{2.9}$$

Now we can write equation (2.7) for all the joints in one vector equation

$$M(\Theta)\ddot{\Theta} + C(\Theta, \dot{\Theta})\dot{\Theta} + N(\Theta, \dot{\Theta}) = \tau \tag{2.10}$$

where τ is the vector of actuator generalized forces, and $N(\Theta, \dot{\Theta})$ includes all the left over generalized forces. In the sequel we shall refer to τ as the vector of joint torques and it will mean exactly the same thing (i.e. actuator applied generalized forces at the joints). This second-order, vector differential equation for the motion of a manipulator as a function of the applied joint torques will be called the *dynamic equation of the manipulator* in the rest of this thesis. We now state without proof the following properties of the matrices M and C which is the reason to derive them in the particular form we have.

Lemma 2.1 (Manipulator dynamic equations: structural properties) *Equation* (2.10) *has the following properties:*

1. $M(\Theta)$ *is symmetric and positive definite.*
2. $\dot{M} - 2C \in \mathbb{R}^{n \times n}$ *is a skew-symmetric matrix.*

The reader is referred to Murray, Li and Sastry [40] for a proof of the above statements. These properties of the dynamic equation are extremely important for their further analysis, and we will have cause to refer to them again when we discuss the stability of control laws for manipulators. Property 2 is sometimes called the *passivity* property of a manipulator. Note that the above properties are true for our derivation of the dynamics, in particular, our choice of the definition of matrix C.

2.1.3 Control of robotic manipulators

In this section we discuss basic robot control methods. We will build on the ideas in this section in more complicated robot control tasks. To begin, we briefly review

some tools used to analyze stability of dynamical systems which will be used extensively throughout the rest of this thesis. These topics are covered in detail in texts like Vidyasagar [58] and Khalil [23] and we will only present the results here.

Stability of dynamical systems

Consider a dynamical system

$$\dot{x} = f(x,t) \qquad x(t_0) = x_0 \qquad x \in \mathbb{R}^n. \tag{2.11}$$

$f(x,t)$ is assumed Lipschitz continuous with respect to x, uniformly in t.

Definition 2.1 (Asymptotic stability)
An equilibrium point x^* (i.e. $f(x^*,t) = 0$) of equation (2.11) is *asymptotically stable* at $t = t_0$ if

1. For any $\epsilon > 0$ there exists a $\delta(t_0, \epsilon) > 0$ such that

$$\|x(t_0) - x^*\| < \delta \qquad \Longrightarrow \qquad \|x(t)\| < \epsilon, \quad \forall t \geq t_0. \tag{2.12}$$

2. There exists $\beta(t_0) > 0$ such that

$$\|x(t_0) - x^*\| < \beta \qquad \Longrightarrow \qquad \lim_{t\to\infty} x(t) = 0. \tag{2.13}$$

Systems satisfying only the first of the above two properties (equation (2.12)) are called *Lyapunov stable*. The above defines stability only at an instant of time t_0. To ensure that the equilibrium point x^* does not lose stability at any time, we require that δ in equation (2.12) and β in equation (2.13) be independent of t_0 so equations (2.12) and (2.13) may hold for all t_0. If this is true the system is said to be *uniformly asymptotically stable*. For autonomous systems (i.e., systems which do not have an explicit time dependence) asymptotic stability is the same as uniform asymptotic stability. Further, the above definition is a *local* definition in that it describes the behavior of a system near an equilibrium point. The equilibrium point x^* is said to be *globally* asymptotically stable if it is asymptotically stable for all initial conditions $x_0 \in \mathbb{R}^n$.

Usually we perform a simple change of coordinates to move the equilibrium point x^* to the origin, that is $x^* = 0$, and then talk about stability around the origin. In the sequel when we talk about the stability of the origin this is exactly what we have done.

Lyapunov's direct method (the second method of Lyapunov) is a general procedure used to determine a systems' stability, without explicit integration of the differential equation. Before we present the theorem we need to state a few definitions. In what follows B_ϵ denotes a ball of size ϵ around the origin, $B_\epsilon = \{x \in \mathbb{R}^n, \|x\| < \epsilon\}$.

Definition 2.2 (Locally positive definite functions)
A continuous function $V : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}$ is *locally positive definite* if for some $\epsilon > 0$

and some continuous, strictly increasing function $\alpha : \mathbb{R}_+ \to \mathbb{R}$,

$$V(0,t) = 0 \quad \text{and} \quad V(x,t) \geq \alpha(\|x\|) \qquad \forall x \in B_\epsilon, \forall t \geq 0.$$

A *positive definite function* is a locally positive function with the additional condition that $\alpha(p) \to \infty$ as $p \to \infty$.

Definition 2.3 (Decresent functions)
A continuous function $V : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}$ is *decresent* if for some $\epsilon > 0$ and some continuous, strictly increasing function $\beta : \mathbb{R}_+ \to \mathbb{R}$,

$$V(x,t) \leq \beta(\|x\|) \qquad \forall x \in B_\epsilon, \forall t \geq 0.$$

We also need to define the following terminology. The time derivative of a scalar continuous function $V(x,t), V : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}$, along the trajectory of the system given by equation (2.11) is the quantity

$$\dot{V}\,|_{\dot{x}=f(x,t)} = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f.$$

We will use $\dot{V}$ to mean $\dot{V}\,|_{\dot{x}=f(x,t)}$ in what follows.

Theorem 2.1 (Lyapunov's theorem for stability) *Consider a non-negative function $V(x,t), V : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}$ with $\dot{V}$ its time derivative along the trajectories of the system given by equation* (2.11)*. Then,*

1. *If V is locally positive definite and $\dot{V} \leq 0$ locally in x for all t, then the origin of the system is locally Lyapunov stable.*
2. *If V is locally positive definite and decrescent, and $-\dot{V}$ is locally positive definite, then the origin of the system is uniformly locally asymptotically stable.*
3. *If V is locally positive definite and decrescent, and $-\dot{V}$ is positive definite, then the origin of the system is globally uniformly asymptotically stable.*

V is called a Lyapunov function for the system.

These are sufficient conditions for the stability of the origin. Though the converse is also true, that is stable systems have Lyapunov functions, as we shall see later, the search for a Lyapunov function is often operose.

The last result we require is a principle which applies to autonomous systems of the form

$$\dot{x} = f(x). \tag{2.14}$$

We denote the solution of this at a time t starting from x_0 at t_0 by $s(t, x_0, t_0)$.

Theorem 2.2 (LaSalle's invariance principle) *Let $V(x), V : \mathbb{R}^n \to \mathbb{R}$ be a locally positive definite function such that on the compact set $\Omega_c = \{x \in \mathbb{R}^n : V(x) \leq$*

$c\}$, $\dot{V}(x) \leq 0$. *Define*

$$S = \{x \in \Omega_c : \dot{V}(x) = 0.$$

Then, as $t \to \infty$, *the trajectory* $s(t, x_0, t_0)$ *tends to the largest invariant set in* S. *In particular, if* S *contains no invariant sets other than* $x = 0$ *then* 0 *is asymptotically stable.*

The use of this principle is to conclude asymptotic stability when the derivative of the Lyapunov function is only negative semi-definite, locally, instead of negative definite.

Controlling basic manipulator tasks

We derived the dynamics of a manipulator in equation (2.10). The elementary robot control problem is to track a given joint trajectory $\Theta_d(t)$ by application of the appropriate actuator forces, τ in equation (2.10). The error in the configuration of the robot is denoted by $e = \Theta_d - \Theta$. The simplest controller which will do the job—if we discount open-loop control laws which will fail unless we have a perfect model of the robot—is the basic *joint level PD* control law given by

$$\tau = K_p e + K_d \dot{e}, \tag{2.15}$$

where K_p and K_d are positive definite matrices. That the PD controller achieves asymptotic set-point ($\dot{\Theta}_d = 0$) stabilization can be proved using

$$V(\Theta, \dot{\Theta}) = \frac{1}{2}\dot{\Theta}^T M(\Theta)\dot{\Theta} + \frac{1}{2}\Theta^T K_p \Theta$$

as a candidate Lyapunov function and then using LaSalle's principle. To achieve tracking the above PD control law needs to be augmented as

$$\tau = M(\Theta)\ddot{\Theta}_d + C(\Theta, \dot{\Theta})\dot{\Theta}_d + N(\Theta, \dot{\Theta}) + K_p e + K_v \dot{e}.$$

The augmented portion of the control law is a variant of the so called a *computed torque* control law. Exponentially stable trajectory tracking can be proved for this controller for $K_p, K_d > 0$ by using the candidate Lyapunov function $V(e, \dot{e}, t) = \frac{1}{2}\dot{e}^T M(\Theta)\dot{e} + \frac{1}{2}e^T K_p e + \epsilon e^T M(\Theta)\dot{e}$, with ϵ sufficiently small.

Control of workspace trajectories

We call the space inhabited by the end-effector of the manipulator, the *workspace* of the manipulator. Let $SE(3)$ denote the special Euclidean group of three-dimensional space. We define

$$g : Q \to SE(3), g(\Theta) = X$$

which maps the configuration of the manipulator to the configuration of the end-effector in workspace coordinates. Q is called the configuration space of the manip-

ulator and $g(\Theta)$ is called the *forward kinematics* of the manipulator. In the most general three-dimensional case $X \in SE(3)$. The forward kinematics can be derived by various different methods. One of the best known uses the *Denavit-Hartenberg parameters* [7]. A more geometric description is the *product of exponentials* derivation of the kinematics [40].

It is natural to prescribe robot trajectories in workspace coordinates. A desired path $g_d(t) \in SE(3)$ prescribes the configuration of the end-effector as a function of time. One way of solving this problem is to solve the inverse kinematics, that is find $\Theta_d(t)$ such that $g(\Theta_d(t)) = g_d(t)$, and then use the methods described previously to achieve tracking in the joint space. However, the inverse kinematics is not very tractable for manipulators with multiple joints. There are multiple solutions to the inverse kinematics problem. A more appealing solution to the problem is to transform the dynamic equations to workspace coordinates. We use local coordinates $\mathbb{R}^p$ instead of $SE(3)$ to parameterize the workspace. (Note that this works only for the fully-actuated non-redundant case, that is when $p = n$, where n is the number of independent actuators in the manipulator.) The forward kinematics, $g : Q \to \mathbb{R}^n, g(\Theta) = X$, is assumed to be a smooth, invertible mapping. The means of transformation is via the *manipulator Jacobian* $J(\Theta)$:

$$\dot{x} = J(\Theta)\dot{\Theta} \qquad J(\Theta) = \frac{\partial g}{\partial \Theta}. \tag{2.16}$$

As we have assumed g smooth and invertible we can write

$$\dot{\Theta} = J^{-1}\dot{X} \qquad \text{and} \qquad \ddot{\Theta} = J^{-1}\ddot{X} + \frac{d}{dt}J^{-1}\dot{X}.$$

Using these we can write the dynamic equations in workspace coordinates

$$\tilde{M}(\Theta)\ddot{X} + \tilde{C}(\Theta, \dot{\Theta})\dot{X} + \tilde{N}(\Theta, \dot{\Theta}) = F, \tag{2.17}$$

where,

$$\begin{aligned} \tilde{M} &= J^{-T}MJ^{-1}, \\ \tilde{C} &= J^{-T}\left(CJ^{-1} + M\frac{d}{dt}(J^{-1})\right)J^{-1}, \\ \tilde{N} &= J^{-T}N, \qquad \text{and,} \\ F &= J^{-T}\tau. \end{aligned}$$

The matrices $\tilde{M}$ and $\tilde{C}$ have the structural properties attributed to the M and C in equation (2.10). (Refer to Lemma 2.1.)

These properties allow us to extend the control laws mentioned previously from the configuration space to the workspace. Given a trajectory $X_d(t)$ in the workspace,

we can use the workspace PD control law

$$F = K_p(X_d - X) + K_d(\dot{X}_d - \dot{X}),$$
$$\tau = J^T F,$$

for set-point regulation. The workspace augmented PD control law

$$F = \tilde{M}(\Theta)\ddot{X}_d + \tilde{C}(\Theta, \dot{\Theta})\dot{X}_d + \tilde{N}(\Theta, \dot{\Theta}) + K_p(X_d - X) + K_d(\dot{X}_d - \dot{X}),$$
$$\tau = J^T F,$$

achieves exponentially stable workspace trajectory tracking.

2.1.4 Modeling constraints in the workspace

Some advanced robotic tasks involve interactions of the manipulator with its environment. Physical interactions of the robot with objects in its environment, including possibly other robots, is usually modeled as a constraint in the robot's workspace. As we shall talk extensively of constrained manipulators in the sequel, we provide a somewhat detailed outline of modeling of workspace constraints within the scope of Lagrangian mechanics.

Before we talk further of workspace constraints we present the following aside on *holonomic constraints* . Holonomic constraints are defined to be those which restrict the motion of the system to a smooth hyper-surface in the configuration space Q. This implies that we can represent a holonomically constrained system using a new, smaller set of unconstrained variables which have the constraints implicitly in them. However, we may not always be able to find a reduced set. We were able to use the joint angles of the manipulator as a reduced set for a free manipulator, but for constraints in the workspace it is not usually possible to find the reduced set of unconstrained coordinates. Locally, we can represent k holonomic constraints as algebraic constraints in the configuration space,

$$h_i(q) = 0, \qquad i = 1, \ldots k, \tag{2.18}$$

where $h_i : Q \to \mathbb{R}$. We assume that the constraints are smooth and linearly independent and hence the matrix

$$\frac{\partial h}{\partial q} = \begin{bmatrix} \frac{\partial h_1}{\partial q_1} & \cdots & \frac{\partial h_1}{\partial q_n} \\ & \ddots & \\ \frac{\partial h_k}{\partial q_1} & \cdots & \frac{\partial h_k}{\partial q_n} \end{bmatrix}$$

is full row rank.

A constraint surface opposes the motion of the system against the constraint. Therefore, an intuitive way of incorporating the effects of the constraint surface is to postulate the existence of forces intrinsic to the surface which oppose motion against the constraint. These are the *constraint forces* we talked about earlier. Constraint surfaces may, in general, have multiple "preferred directions." Since we need to

define the constraint force direction for all surfaces, and planes (or hyper-planes) and spherical surfaces have the surface normal as the unique preferred direction, the direction of the gradient of the surface is taken to be the direction of the constraint force. For a single scalar constraint $l(q) = 0$ the constraint force is given by

$$f_{lc} = (\nabla l)\lambda.$$

The gradient sets the direction of the constraint. The undetermined scalar factor λ sets the magnitude of the constraint force and is called a *Lagrange multiplier.* For the set of holonomic constraints, $h_i(q) = 0, i = 1, \ldots, k$, the constraint force is a linear combination of the forces due to each constraint,

$$F_{hc} = \frac{\partial h}{\partial q}^T \lambda,$$

where $\lambda \in \mathbb{R}^k$ is the vector of the Lagrange multipliers. Note that this setup for constraint forces is consistent with D'Alembert's principle and no work is done by the constraint forces. Also note the following property which will be extremely useful in the sequel:

$$\dot{h} = 0 = \frac{dh}{dt} = \frac{\partial h}{\partial q}\dot{q}. \tag{2.19}$$

As mentioned earlier (refer equation (2.1)) we can write a set of constraints, more generally, in the Pfaffian form

$$A(q)\dot{q} = 0,$$

(the constraints are assumed to independent of time here), where $A(q) \in \mathbb{R}^{k \times n}$ represents a set of k velocity constraints. The Pfaffian form of the holonomic constraints $h_i(q) = 0, i = 1, \ldots k$, is $\frac{\partial h}{\partial q}\dot{q} = 0$. Nonholonomic constraints can also be represented as Pfaffian constraints, but they cannot be integrated to obtain algebraic constraints in the configuration space. We noted that holonomic constraints implicitly satisfy D'Alembert's principle. Constraint forces for nonholonomic constraints which satisfy D'Alembert's principle can be written identically to those for holonomic constraints:

$$F_A = A^T(q)\lambda,$$

where $\lambda \in \mathbb{R}^k$ is, as before, the vector of Lagrange multipliers.

We can incorporate smooth, linearly independent constraints, written in the Pfaffian form

$$A(q)\dot{q} = 0, \qquad A(q) \in \mathbb{R}^{k \times n} \tag{2.20}$$

by considering the constraint forces as an additional force affecting the motion of the system. Adding these forces to Lagrange's equations (2.5) we get the constrained

dynamics

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} + A^T(q)\lambda = Q_e. \tag{2.21}$$

The Lagrange multipliers are determined by simultaneously solving equations (2.20) and (2.21) for the $n+k$ variables q and λ. This guarantees that there will be no motion in the constrained directions.

In case of a robotic manipulator, we can obtain an explicit formula for the calculation of λ as follows. The dynamics (equation (2.10)) with constraints $A(\Theta)\dot{\Theta} = 0, A(\Theta) \in \mathbb{R}^{k\times n}$ embedded in it can be written

$$M(\Theta)\ddot{\Theta} + C(\Theta,\dot{\Theta})\dot{\Theta} + N(\Theta,\dot{\Theta}) + A^T\dot{\Theta} = \tau. \tag{2.22}$$

The constraint equation in the Pfaffian form can be differentiated to obtain

$$A(\Theta)\ddot{\Theta} + \dot{A}(\Theta)\dot{\Theta} = 0.$$

Substituting $\ddot{\Theta}$ from equation (2.22) into the above equation and rearranging terms we get for λ

$$\lambda = (AM^{-1}A^T)^{-1}\left(AM^{-1}(\tau - C\dot{\Theta} - N) + \dot{A}\dot{\Theta}\right). \tag{2.23}$$

The Lagrange multipliers can now be computed as a function of the current state, Θ and $\dot{\Theta}$, of the manipulator and the applied external torque τ. Once computed these can be substituted into the equation (2.22) to compute the motion of the system.

2.1.5 Hybrid force-position control

A question which naturally arises from the above discussion is that of control of forces of constraint. In applications which involve constrained manipulators, it is very likely that the force of interaction (against the constraint surface) needs to be regulated in addition to the position of the manipulator "along" the constraint surface. This is the problem addressed by *hybrid force-position control.* There are quite a few technical difficulties in not only trying to synthesise hybrid control methods and proving their stability, but even in posing the problem, globally, in a universally acceptable framework.

We now describe a hybrid control methodology for manipulators. In what follows we assume a local, Euclidean coordinate representation of the workspace. We also assume that the constraints are holonomic. Therefore, if there are k constraints $h_i(\Theta) = 0, i = 1,\ldots,k$, we can use a set of $n-k$ coordinates, say, $\Phi = (\phi_1,\ldots,\phi_{n-k})$ to parameterize the constraints. There is a smooth injective map $f : \mathbb{R}^{n-k} \to \mathbb{R}^n$ such that

$$h_i(f(\Phi)) = 0.$$

Letting $J = \frac{\partial f}{\partial \Phi}^{-1}$ we can rewrite the dynamics exactly as in equation (2.17) with

the X replaced by Φ:

$$\tilde{M}(\Theta)\ddot{\Phi} + \tilde{C}(\Theta, \dot{\Theta})\dot{\Phi} + \tilde{N}(\Theta, \dot{\Theta}) = F.$$

We are assuming that the manipulator remains in contact with the constraint at all times. Consider a path on the constraint surface given by $\Phi_d(t)$ and a normal force to be applied to it specified by the Lagrange multipliers $\lambda_1(t), \ldots, \lambda_k(t)$. A controller which achieves position control of the manipulator is

$$\tau_\Phi = J^T \left(\tilde{M}(\Theta)(\ddot{\Phi}_d + K_p e_\Phi + K_d \dot{e}_\Phi) + \tilde{C}(\Theta, \dot{\Theta})\dot{\Phi} + \tilde{N}(\Theta, \dot{\Theta}) \right),$$

with $e_\Phi = \Phi_d - \Phi$. This control moves the manipulator so it tracks the correct trajectory on the constraint *without applying any force against it.* This means if the manipulator is started off with its end-effector touching the constraint surface, it will track the required trajectory, without pushing against the constraint. To make the manipulator apply the constraint force we add on the torque required for the normal force

$$\tau_N = \sum_{i=1}^{k} \lambda_i(t) \nabla h_i.$$

The full control law is therefore

$$\tau = \tau_\Phi + \tau_N.$$

We will discuss more general cases of control of constrained manipulators when we talk about controllers for grasping.

2.2 Grasping Kinematics, Dynamics and Control

Grasping with robotic manipulators is an application where multiple robots interact with each other. Each finger in a grasping setup is a manipulator. Seeing that this work is motivated, partially, by applications of robotic grasping, and that there is a large amount of experimental data we present from a grasping setup, we describe in this section how we extend our tools to the study and control of grasping. Note that most of what we describe applies to any setup with interacting robots and is not specific to grasping. Our development of grasping follows that presented in Murray, Li and Sastry [40], and the reader is referred to that publication for details.

2.2.1 Robotic hand kinematics

To be able to grasp with robotic fingers (each of which is a manipulator in its own right), we need to be able to find out the relationships between the forces and motions of the whole finger-object system. We assume as given the models of the robotic fingers, the object and a description of the contact points as well as the nature of the contacts themselves. The desirable properties of a grasp include:

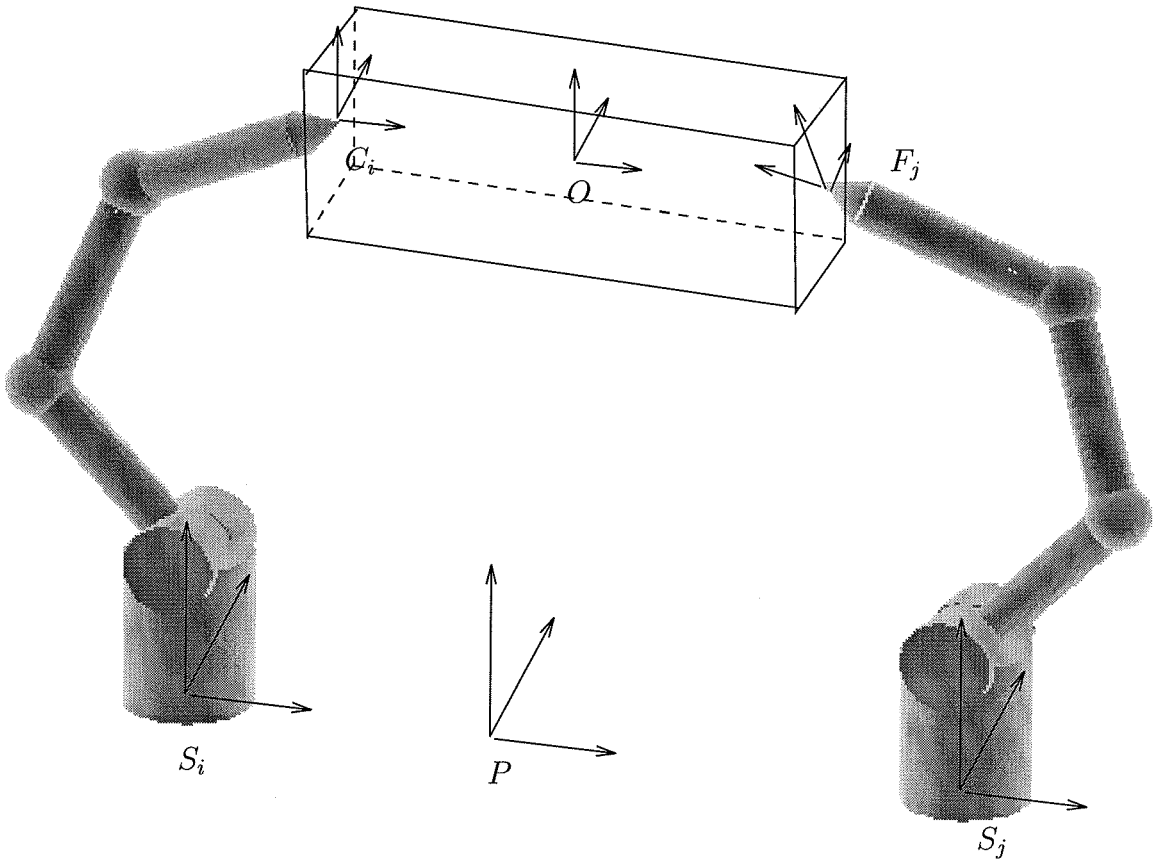


Figure 2.2 Coordinate frames for grasping.

1. The ability to resist external forces.
2. The ability to manipulate the object.

Determination of a set of contact points satisfying these criteria, is the problem of *grasp planning*, and is a field of research in itself. Having noted this, we will always assume in the sequel that the planning process has been accomplished.

The kinematics equations for grasping are derived under the assumption that the finger never lose contact with the object being manipulated at the point of contact. Equivalently, we can require that the relative velocity between the tip of each finger and the point of contact of the finger with the object be zero at every contact point. Before going further we describe the following coordinate frames for grasping (refer to Figure 2.2).

Palm frame: All the fingers of the "robot hand" are attached to a common base—"the palm." The palm frame, P, is attached rigidly to the palm.

Finger base frame: Each finger has a frame, S_i, attached to its base. This frame is stationary with respect to the palm frame.

Finger tip frame: Each finger has a frame, F_i, attached to its tip. This finger moves with the tip of the finger.

Contact frame: Each contact has a frame, C_i, which has its origin at the point of contact and is attached to (and moves with) the grasped object. As a matter of convention, the inward pointing normal to the contact surface, at the point of contact will be along the y axis of the contact frame.

Object frame: The object has a frame, O attached rigidly to it. This is the object frame.

We shall find the following two constructs very useful for describing grasp kinematics and dynamics:

The grasp map G: The grasp map is used to transform the forces applied at each contact of the object, in the contact frame, to the resulting wrench on the grasped object in the body frame. If the object is in a p-dimensional space, and m_i independent forces/torques can be applied at the ith contact point, then we can use $G_i \in \mathbb{R}^{p \times m_i}$, a linear map to transform the contact force, f_{c_i}, at the ith contact point to the body frame, i.e.

$$F_{o_i} = G_i f_{c_i}.$$

Define for a grasp with k fingers, $G : \mathbb{R}^m \to \mathbb{R}^p$, $m = m_1 + \cdots + m_k$,

$$G = \begin{bmatrix} G_1 & G_2 & \cdots & G_k \end{bmatrix}$$

and

$$f_c = \begin{bmatrix} f_{c_1} \\ f_{c_2} \\ \vdots \\ f_{c_k} \end{bmatrix}.$$

As the wrench mapping is linear we can add together the wrenches at the object frame for each contact force to get the total wrench at the body frame as

$$F_o = G f_c. \tag{2.24}$$

Given the above we can deduce from the principle of conservation of work the following relation between the velocity of the object in the object frame, V_o (its body velocity) and the velocity of each contact frame with respect to the contact frame, $\dot{X}_{c_i}$:

$$\dot{X}_c = \begin{bmatrix} \dot{X}_1 \\ \vdots \\ \dot{X}_k \end{bmatrix} = G^T V_o. \tag{2.25}$$

The hand Jacobian J_h: The hand Jacobian is used to transform joint velocities

of the finger to the velocities of the tip frames *in the contact frame*. If $\tilde{J}_i$ represents the mapping of the joint velocities to the tip velocity in the contact frame for each finger, then,

$$J_h = \begin{bmatrix} \tilde{J}_1 & & 0 \\ & \ddots & \\ 0 & & \tilde{J}_k \end{bmatrix}. \tag{2.26}$$

Note that $\tilde{J}_i$ are different from the usual manipulator Jacobian, because we are representing the tip velocity in the contact frame and not the finger base frame as would be the case for the usual manipulator Jacobian. With

$$\Theta = \begin{bmatrix} \Theta_1 \\ \vdots \\ \Theta_k \end{bmatrix},$$

we can write

$$\dot{X}_c^* = J_h \dot{\Theta}, \tag{2.27}$$

for the velocity of the finger tips in their respective contact frames. The force relation that holds is

$$\tau = \begin{bmatrix} \tau_1 \\ \vdots \\ \tau_k \end{bmatrix} = J_h^T f_c, \tag{2.28}$$

where τ_i are the joint torques for the ith finger.

Now we can state the grasping kinematics in terms of the *fundamental grasping constraint* as

$$J_h(\Theta, X_o)\dot{\Theta} = G^T V_o. \tag{2.29}$$

Here X_o is the position of the object. The above equation (2.29) is just the mathematical expression for the assumption stated at the beginning of this section.

To clarify the kinematics further we present an example here for the case of planar grasping. This example resembles our experimental grasping setup.

Example 2.1 (Grasping kinematics for a two-finger planar hand)
The finger setup is as shown in Figure 2.3. We derive the kinematic equations for grasping for this two-dimensional setup. The contact points are assumed to be non-slipping, point contacts with friction. Therefore at each contact point the manipulator can apply forces normal to and parallel to the surface of contact, while staying within the friction cone. We use $V_{op} \in \mathbb{R}^3$ to denote the velocity of the object frame with respect to the palm frame. V_o has as its elements the linear velocities in the x and y directions and the angular velocity about the z direction respectively.

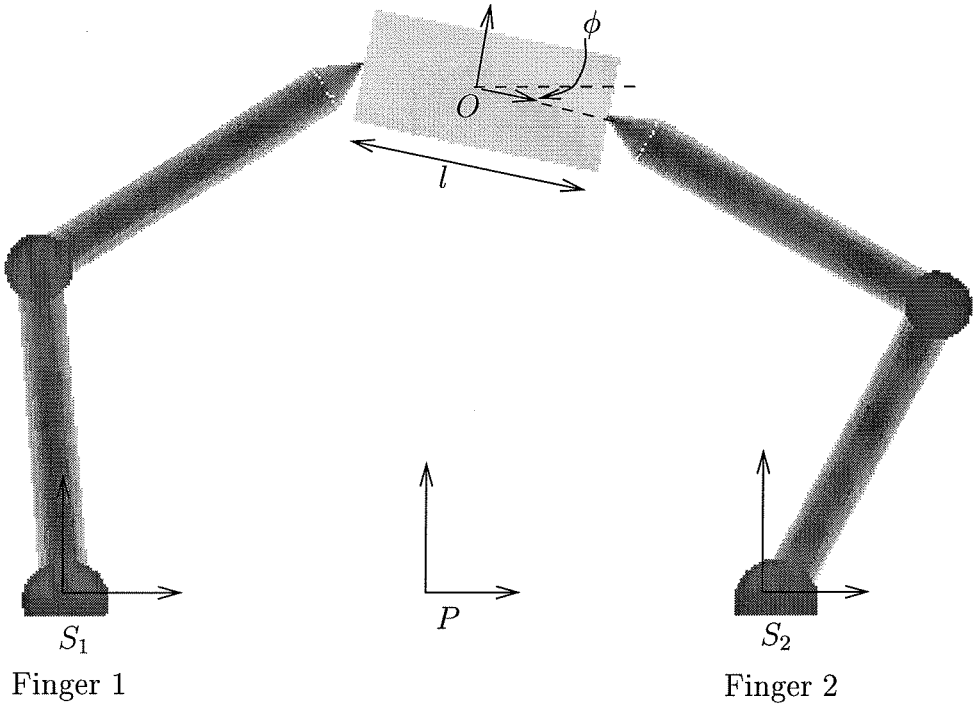


Figure 2.3 A planar, two-finger grasping setup.

Now we can write for the velocity of the object in the object frame

$$V_o = \begin{bmatrix} R_\phi & 0 \\ 0 & 1 \end{bmatrix}^{-1} V_{op}$$

with the notation that R_α denotes the rotation matrix for an angle α about the z axis perpendicular to the xy plane:

$$R_\alpha = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}.$$

The velocity of the points of contact can now be written as

$$\dot{X}_c = G^T V_o = G^T \mathrm{Ad}_\phi^{-1} V_{op},$$

where we have denoted by Ad_ϕ the quantity $\begin{bmatrix} R_\phi & 0 \\ 0 & 1 \end{bmatrix}$ and G is the grasp map which

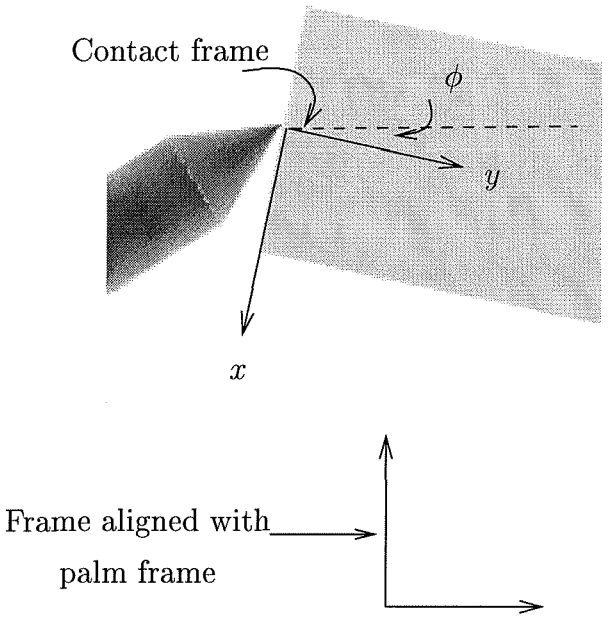


Figure 2.4 Contact point magnified.

for our setup (see Figure 2.3) is

$$G = \begin{bmatrix} 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ l/2 & 0 & l/2 & 0 \end{bmatrix}.$$

Its transpose maps the body velocity of the grasped object V_o to the velocities of the points of contact *in the contact frame.* It is useful to get the velocities in the contact frame because the null forces—that is forces resulting in no motion of the object—are always along the y-axes of the contact frames (see Figure 2.4). In the palm frame the tip velocity of each finger is given by

$$\dot{X}_i = J_i \dot{\Theta}_i,$$

with J_i the usual manipulator Jacobian of the ith robot. In the contact frame these velocities can be written as

$$\dot{X}_{c_i} = R_{c_i}^{-1} J_i \dot{\Theta}_i,$$

where R_{c_i} is the rotation matrix between the frame aligned with the palm frame and the contact frame at the point of contact. For our setup

$$R_{c_1} = R_{\phi - \frac{\pi}{2}}$$

and

$$R_{c_2} = R_{\phi+\frac{\pi}{2}}.$$

The kinematic equations are given by equating the velocities:

$$\begin{bmatrix} R_{c_1}^{-1} J_1 & 0 \\ 0 & R_{c_2}^{-1} J_2 \end{bmatrix} \dot{\Theta} = G^T \mathrm{Ad}_{\phi}^{-1} V_{op},$$

or in the notation of the previous section,

$$J_h \dot{\Theta} = \begin{bmatrix} \tilde{J}_1 & 0 \\ 0 & \tilde{J}_2 \end{bmatrix} \dot{\Theta} = G^T V_o,$$

where $\tilde{J}_i = R_{c_i}^{-1} J_i$.

We point out here that the above example is one of the simplest possible. The equations for three-dimensional grasping, with multiple fingers and rolling allowed at the points of contact would be vastly more complicated.

2.2.2 Robot hand dynamics

The dynamics for a robot hand grasping an object are obtained by combining together the dynamic equations of the fingers and the object. We write the dynamics of the ith finger (refer equation (2.10)) as

$$M_i(\Theta_i)\ddot{\Theta}_i + C_i(\Theta_i, \dot{\Theta}_i)\dot{\Theta}_i + N_i(\Theta_i, \dot{\Theta}_i) = \tau_i.$$

Combining the equations together, we can write for the entire hand with k fingers

$$M_f(\Theta_f)\ddot{\Theta}_f + C_f(\Theta_f, \dot{\Theta}_f)\dot{\Theta}_f + N_f(\Theta_f, \dot{\Theta}_f) = \tau_f, \tag{2.30}$$

where

$$M_f = \begin{bmatrix} M_1 & & 0 \\ & \ddots & \\ 0 & & M_k \end{bmatrix}; C_f = \begin{bmatrix} C_1 & & 0 \\ & \ddots & \\ 0 & & C_k \end{bmatrix}; N_f = \begin{bmatrix} N_1 \\ \vdots \\ N_k \end{bmatrix}; \tau_f = \begin{bmatrix} \tau_1 \\ \vdots \\ \tau_k \end{bmatrix}; \Theta_f = \begin{bmatrix} \Theta_1 \\ \vdots \\ \Theta_k \end{bmatrix}.$$

We write the dynamics of the object in local coordinates as

$$M_o(X)\ddot{X} + C_o(X, \dot{X})\dot{X} + N_o(X, \dot{X}) = 0, \tag{2.31}$$

with $X \in \mathbb{R}^6$, a set of local coordinates for $SE(3)$. Elements of this equation satisfy the same structural properties as those in equation (2.10). Additionally we have the grasping constraint (equation (2.29)):

$$J_h(\Theta, X_o)\dot{\Theta} = G^T V_o.$$

We need to make three assumptions about the grasp to derive its dynamics:

1. *The grasp is force closure and manipulable.* Force closure implies that the grasp is able to resist all applied wrenches to the object (assuming that the fingers have no limits on the forces they can apply). Mathematically this is equivalent to stating that given any external wrench $F_e \in \mathbb{R}^p$ on the object, we have a $f_c \in FC$, (FC denotes the friction cone) such that

$$Gf_c = -F_e.$$

   Manipulability implies the ability of the fingers to follow any object motion while grasping it. Mathematically, this can be stated as $\mathcal{R}(G) \subset \mathcal{R}(J_h)$, where $\mathcal{R}(\cdot)$ denotes the range of the mapping.

2. *The hand Jacobian is invertible.* This ensures that we have no redundant degrees of freedom.

3. *The contact forces remain in the friction cone at all times.* This ensures that the grasp constraints are always satisfied.

Under these conditions and letting $q \in \mathbb{R}^n \times \mathbb{R}^p$ represent the variables (Θ, X), we get the dynamic equations of the grasp (refer to [40] for details) as

$$M_G(q)\ddot{X} + C_G(q,\dot{q})\dot{X} + N_G(q,\dot{q}) = F, \tag{2.32}$$

where

$$\begin{aligned} M_G &= M_o + GJ_h^{-T} M_f J_h^{-1} G^T \\ C_G &= C_o + GJ_h^{-T}\left(C_f J_h^{-1} G^T + M_f \frac{d}{dt} J_h^{-1} G^T\right) \\ N_G &= N_o + GJ_h^{-T} N_f \\ F &= GJ_h^{-T}\tau_G. \end{aligned}$$

These equations have the same form as the equations for a single open-chain manipulator in equation (2.10) and satisfy the same structural properties.

2.2.3 Control of robot hands

Controlling a grasp is an exercise in hybrid force-position control. We are required to control the trajectory of the object and in addition maintain a desired *internal force.* An internal force in a grasp is a force which by itself does not cause any motion of the object. In one sense the internal force determines "how hard" the object is being grasped. Mathematically, a contact force, f_N is an internal or null force if $f_N \in \mathcal{N}(G)$, where $\mathcal{N}(\cdot)$ denotes the null space of a linear operator.

To achieve control we first compute the forces required to move the body towards the desired trajectory. Given a trajectory X_d the required object wrench required for a computed torque control law would be

$$F = M_G(q)(\ddot{X}_d + K_p e + K_d \dot{e}) + C_G(q,\dot{q})\dot{X}_d + N_G(q,\dot{q}).$$

The proof of asymptotic convergence to the desired trajectory follows directly from those for the simple open-chain manipulator case. The required joint torques can now be chosen to satisfy

$$GJ_h^{-T} = F.$$

Note that because of our assumptions, the map GJ_h^{-T} is surjective and therefore the above equation always has a solution. Indeed there are multiple solutions, which reflect the existence of the internal forces. The general solution assuming J_h invertible is therefore

$$\tau_G = J_h^T G^+ F + J_h^T f_N, \tag{2.33}$$

where $G^+ = G^T(GG^T)^{-1}$ is the pseudo-inverse of G and $f_N \in \mathcal{N}(G)$. We can select f_N to satisfy our internal force requirements without affecting the trajectory tracking, because G annihilates these forces. Hence, we can use the control law

$$\tau_G = J_h^T G^+ F + J_h^T f_{N_d},$$

where f_{N_d} is the desired internal force. Note that the dynamics of the system will tend to change the real internal force experienced by the object. This is a hard problem to solve in general and we assume that the internal force is high enough that the other forces are small compared to it. Sensing of the finger tip forces and a feedback adjustment of the forces is also a possibility, but we must be careful to not get into algebraic loops doing this. There is also the distinction between internal forces and "squeezing forces" [28] which we will not discuss here.

2.3 Structural Flexibility in Robotic Manipulators

Incorporation of structural flexibility into the kinematic and dynamic equations of robotic manipulators is a subject of ongoing research. In this section we present an overview of some ideas proposed for introducing flexibility into the manipulator equations.

Structural flexibility can manifest itself in manipulators in two forms: *joint flexibility* and *link flexibility*. Joint flexibility is usually easier to model and control because it is localized at a joint of the manipulator. Link flexibility is rather harder to deal with. We will only discuss link flexibility in this section. Fraser and Daniel [9] is a good reference for the material in this section.

2.3.1 Link flexibility

The flexible behavior of links is modeled by the theory of flexible beams [10]. Bernoulli's law for bending beams states that bending moment at any point of a beam is proportional to the change in curvature caused at that point by action of the load. If $\mathcal{M}$ is the bending moment and r the radius of curvature at a point

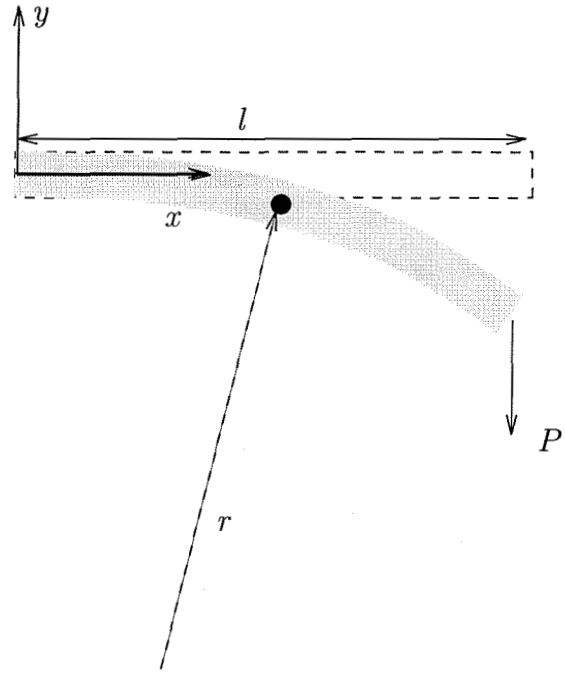


Figure 2.5 Cantilever under load.

(refer to Figure 2.5) then we can write

$$\mathcal{M} = \frac{C}{r},$$

with C a constant. In the Cartesian coordinates shown, the radius of curvature r is given by

$$r = \frac{\left(1 + (\frac{dy}{dx})^2\right)^{\frac{3}{2}}}{\frac{d^2y}{dx^2}}.$$

The full bending equation therefore becomes

$$\mathcal{M} = \frac{C\frac{d^2y}{dx^2}}{\left(1 + (\frac{dy}{dx})^2\right)^{\frac{3}{2}}}. \tag{2.34}$$

This equation is called the *Bernoulli-Euler beam equation.* This equation is the basis of analysis of deflection of planar beams. It is a second order nonlinear differential equation which cannot be solved in general. In engineering applications equation (2.34) is linearized by neglecting the $(dy/dx)^2$ term in the denominator. However, this works well only for deflections which are small in comparison to the

length of the beam. Further, the above equations are for a statically loaded beam. When the loading is varied the equations become vastly more complicated. We have to solve fourth-order partial differential equations, which for the linear case are of the form

$$C\frac{\partial^4 y}{\partial x^4} + m\frac{\partial^2 y}{\partial t^2} = 0, \tag{2.35}$$

with boundary conditions dependent on the loading and constraints at the ends of the beam.

2.3.2 Modeling alternatives

As should be evident from the foregoing discussion a closed form solution for the flexibility is not realizable. Equation (2.35) can be solved by separation of variables giving individual solutions of the form

$$y_i(x,t) = \phi_i(x)\eta_i(t), \tag{2.36}$$

with η_i purely a function of time (and includes an arbitrary constant) and ϕ_i purely a function of the displacement along the beam. The ϕ_i are called the *mode shapes* of the beam and the η_i are the time-dependent amplitudes. The full solution is the infinite sum

$$\begin{aligned} y(x,t) &= \sum_{i=0}^{\infty} y_i(x,t) \\ &= \sum_{i=0}^{\infty} \phi_i(x)\eta_i(t). \end{aligned} \tag{2.37}$$

The zeroth mode is the rigid body mode.

Once the mode shapes and the time dependent amplitude functions have been determined, we can use these to derive the dynamics by introducing the energy due to the flexibility into the Lagrangian. The kinetic energy is given by (refer to Figure 2.5)

$$T_{fl} = \frac{1}{2}\int_0^l \dot{y}(x,t)^2 m dx, \tag{2.38}$$

where m is the mass per unit length of the beam. The potential energy is given by

$$V_{fl} = \frac{1}{2}\int_0^l C y''(x,t)^2 dx, \tag{2.39}$$

where we have denoted differentiation with respect to x by $'$. Using orthogonality properties of the modes and substituting in the solution for y from equation (2.37)

the kinetic and potential energies of the system can be written as

$$T_{fl} = \frac{1}{2}\sum_{i=1}^{\infty} I_{fl}\dot{\eta}_i^2 \tag{2.40}$$

$$V_{fl} = \frac{1}{2}\sum_{i=1}^{\infty} I_{fl}\omega_i^2\eta_i^2, \tag{2.41}$$

where we have used

$$\int_0^l m\phi_i(x)\phi_j(x)dx = I_{fl}\delta_{ij},$$

and δ_{ij} is the Kronecker delta. The η_i form a set of generalized coordinates for a Lagrangian analysis of the system. To make the problem manageable a truncation of the number of modes is carried out and only the first few modes (usually $\leq$ 10) are used. This is one of the most widely used methods used for modeling flexibility. It is also called the *assumed modes* method for modeling flexibility. Flexible manipulators can be adequately modeled with this technique using a finite number of modes. The number of modes required depends on the frequencies of interest and the performance goal for the manipulator. In general finding the mode shapes and the time dependent amplitudes is a nontrivial endeavor, especially in the case of multi-link, constrained manipulators where the boundary conditions can be fairly complicated.

Another way of modeling flexibility within the Lagrangian setup is to use finite elements. A set of displacement and/or slope values at certain points on the flexible beam (nodes) are used as generalized coordinates and the shape of the beam in between these is given by shape functions dependent on x. Expressions for kinetic and potential energies of the system are developed from these. Usoro et al. [56] use this approach for modeling of flexible manipulators.

2.3.3 The rigid sub-link model

We wish to consider deflections in excess of those to which the linear model applies, but the full nonlinear beam model is not very tractable. The assumed modes method is difficult to extend to multiple links because of complicated boundary conditions. The finite element based methods are usually computationally intensive. Therefore, for our analysis and simulation, we choose to use a finite link model for the flexible link. This model replaces a flexible link with a series of rigid sub-links connected through linear torsional springs and dampers, such that the lengths of the sub-links add up to that of the original link (see Figure 2.6). With appropriate values for the spring constants this model can estimate the actual modes of a flexible beam. Larger numbers of sub-links improve the approximation.

In contrast to the assumed modes methods, the modeling of links as a series of masses and springs does not require the *a priori* assumption of mode shapes. This has the advantage that the model parameters can be identified and verified from

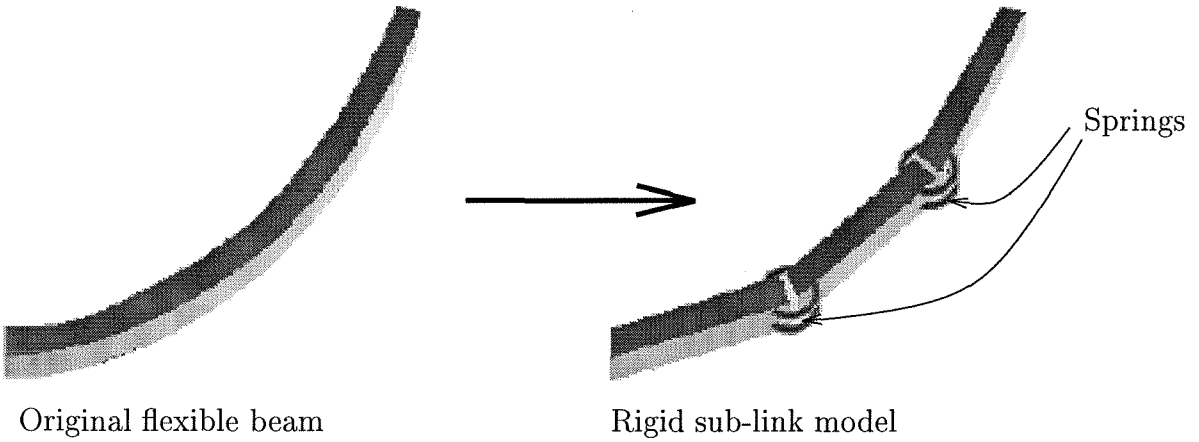


Figure 2.6 The rigid sub-link model of flexibility.

experimental data and the model can be tuned to agree closely with the actual robot link. The number and stiffness values of the springs can be selected appropriately to represent a realistic model. Zaki and El Maraghi in [60] state for a cantilever beam, that treating the links as Bernoulli-Euler beams and matching the end point deflection of the actual beam to the discrete link approximation the, appropriate spring stiffness to use in the sublink model is given by

$$k_t = \frac{3EI}{L}\frac{(n-1)(2n-1)}{6n},$$

where n is the number of sublinks in the model and EI and L are the flexural rigidity and the length of the continuous model respectively. Note that the expression for the spring constant grows unboundedly as the number of sublinks is increased. Representing a cantilever beam with three segments the first mode of the beam could be estimated correctly. However, it was found that the estimation of the second mode was below the actual second mode of the beam [60]. The number of sub-links required to get good estimates for higher modes would of course be larger.

Yoshikawa and Hosoda in [59] present one technique of getting the model parameters for an actual robot manipulator and verify its accuracy against a real experiment. They derive the dynamic model of the beam using virtual rigid links and passive joints consisting of springs and dampers. The parameters of the virtual links and passive joints are identified from measured data of the real flexible link.

Instead of basing the model on just one characteristic they select several static (e.g., deformation of the tip under load) and dynamic (e.g., natural frequency) characteristics of the real link and measure their values. The parameters of the model are then tuned to minimize the weighted error between the characteristics of the real link and the model. Denoting the characteristics of the real arm by α_r and

those of the model by α_m, parameters for the model are selected to minimize

$$J = \sum_i w_i \frac{(\alpha_{r_i} - \alpha_{m_i})^2}{\alpha_{r_i}^2},$$

where w_i is the weight accorded to each characteristic. The characteristics for the model can be calculated quite easily from the dynamic equations of the sublink model.

For example, in Figure 2.6 if we consider the link to be a planar cantilever with no joint flexibility (i.e., the first link is fixed), and disregard damping the dynamic equation is

$$\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} \ddot{\phi}_1 \\ \ddot{\phi}_2 \end{bmatrix} + \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} = 0,$$

with ϕ_1, ϕ_2 the angles at the passive joints, the m the inertia terms and the k the springs constants at the joints. Note that $m_{12} = m_{21}$. Disregarding second and higher order terms of vibration, the first and second natural frequencies of the sublink model are the solutions of the equation

$$k_1 k_2 - \omega^2(m_{22}k_1 + m_{11}k_2) + \omega^4(m_{11}m_{22} - m_{12}^2) = 0.$$

The static deformations at the tip due to the action of forces or moments are determined as follows. The angles ϕ_1, ϕ_2 at the passive joints are solutions of the equations

$$\begin{bmatrix} k_1\phi_1 \\ k_2\phi_2 \end{bmatrix} = \begin{bmatrix} -l_1 \sin\phi_1 - l_2 \sin(\phi_1 + \phi_2) & l_1 \cos\phi_1 + l_2 \cos(\phi_1 + \phi_2) & 1 \\ -l_2 \sin(\phi_1 + \phi_2) & l_2 \cos(\phi_1 + \phi_2) & 1 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ M \end{bmatrix}$$

where l_0, l_1, l_2 are the lengths of the three sublinks and P_x, P_y, M are the forces in the x and y directions and the moment respectively. The static linear and angular deflection of the tip are given by

$$\begin{bmatrix} u_x \\ u_y \\ \phi_m \end{bmatrix} = \begin{bmatrix} l_1 \cos\phi_1 + l_2 \cos(\phi_1 + \phi_2) - l_1 - l_2 \\ l_1 \sin\phi_1 + l_2 \sin(\phi_1 + \phi_2) \\ \phi_1 + \phi_2 \end{bmatrix},$$

where u_x and u_y are deflections in the x and y directions respectively and ϕ_m is the angular deflection.

The incorporation of the rigid sub-link approach into the Lagrangian setup is extremely straightforward. Once the parameters of the sublink model for each flexible link are computed these links are connected for the full manipulator. The original flexible manipulator is replaced by a rigid manipulator with greater number of links and with passive, unactuated joints. The dynamic equations follow exactly as shown before for the rigid case. The extra parameters introduced are the angles at the unactuated joints. These serve as additional generalized coordinates in the

equations for flexible manipulators. The form of the equations for a free manipulator (compare with equation (2.10)) is

$$\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} \ddot{\Theta} \\ \ddot{\Psi} \end{bmatrix} + \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & k_s \end{bmatrix} \begin{bmatrix} \Theta \\ \Psi \end{bmatrix} + \begin{bmatrix} k_{f_1} & 0 \\ 0 & k_{f_2} \end{bmatrix} \begin{bmatrix} \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} \tau \\ 0 \end{bmatrix}. \tag{2.42}$$

The Θ are the angles at the actuated joints (the real joints of the robot) and the Ψ are the joint angles for the passive joints from the sublink model. The spring constants for the sublink model are given by k_s and the damping at the joints are modeled by k_{f_1} and k_{f_2}. We have not considered any other nonlinear effects. Note that on the right hand side of the equations there are no external torques applied directly to the passive joints as they cannot be directly actuated. The structure of these equations is important for our analysis and we shall have occasion to refer to them in the sequel.

We present an example here, which again is from our experimental setup, to clarify the dynamic modeling using the rigid sub-link model.

Example 2.2 (Dynamics of a manipulator, with the last link flexible) The example system is shown in Figure 2.7. The manipulator is a planar, two-degree of freedom, revolute jointed robot with the last link flexible. In what follows $q \in \mathbb{R}^n$ is used to denote the vector of all joint angles, Θ the vector of actuated joint angles and Ψ the vector of unactuated joint angles (refer to Figure 2.7), i.e.,

$$q = \begin{bmatrix} \Theta \\ \Psi \end{bmatrix} \qquad \Theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \qquad \Psi = \begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix}.$$

The usual manipulator dynamics equation is obtained by carrying out the procedure outlined previously

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + N(q,\dot{q}) = \tilde{\tau},$$

with,

- M the inertia matrix,
- C the Coriolis matrix,
- $\tilde{\tau}$ the applied joint torques, and
- N the vector of other generalized forces.

The vector N is split into two components, one of which is related to the springs on the passive joints and the other to the damping. We assume a viscoelastic model for the beam flexibility. The equation obtained is therefore,

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + K_s q + K_f \dot{q} = \begin{bmatrix} \tau \\ 0 \end{bmatrix},$$

where the bottom zeroes in the torque vector are due to the unactuated joints. The matrix K_s is a diagonal matrix of spring constants with the elements corresponding

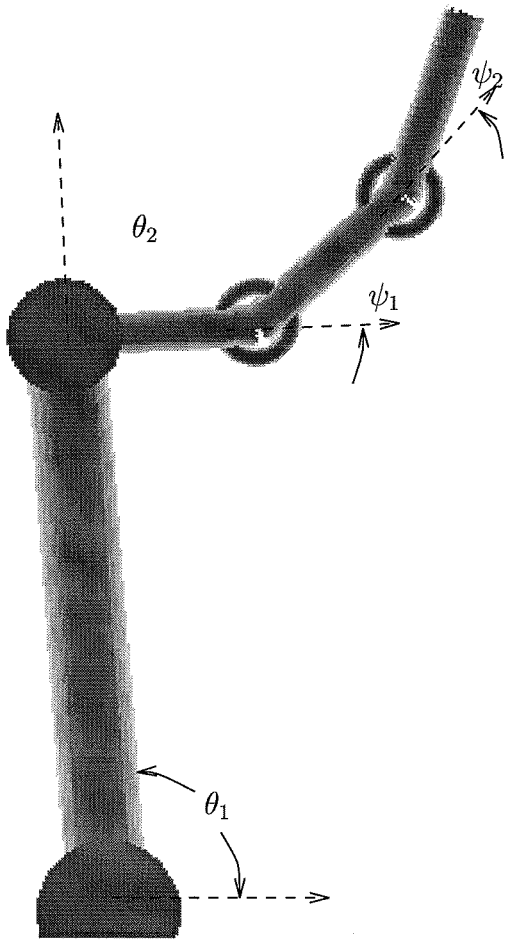


Figure 2.7 Manipulator with last link flexible.

to the actuated joints (without springs) equal to zero:

$$K_s = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & k_{s_1} & 0 \\ 0 & 0 & 0 & k_{s_2} \end{bmatrix}.$$

K_f is similar to K_s but with diagonal elements corresponding to the damping coefficients. K_f can also hold the values of frictional damping coefficients for the actuated joints.

Chapter 3

Singular Perturbation Analysis

Singular perturbation analysis is often used as a means of reducing the order of a system by using its time-scale properties, a generalization of the concept of invariant subspaces in linear systems to nonlinear systems. For control purposes singular perturbation methods are both, a set of tools for modeling and a framework for controller design. However their key contribution is in the realm of modeling and model simplification.

In this chapter we discuss the basis of singular perturbation analysis. The standard form of the mathematical exposition of the theory of singular perturbations is presented next. We then extend and modify the theory to a form most convenient for our analysis. The reader is referred to the publications of Tikhonov [53, 54] and Hoppensteadt [16, 17, 18] and the book by Khalil [23] for greater details on the standard singular perturbation theory. A review of singular perturbations in control problems is provided by Kokotovic [24].

3.1 Standard Singular Perturbation Analysis

In the systems analysis and modeling literature, singular perturbation theory is generally presented for a system of first-order ODEs. Though we can convert the dynamic equations of the robot to a system of first-order ODEs, it is attractive to preserve their second-order form for intuitive clarity. In this section we present the theory in its first order form first. A discussion of the time-scale properties of the system follows. We then present a modification of the approach for second-order ODEs.

3.1.1 The standard model

In the first order presentation of singular perturbation theory, the *standard singular perturbation model* denotes the system of (possibly vector) dynamic equations

$$\begin{aligned} \dot{x} &= f(t,x,y,\epsilon), \qquad x \in \mathbb{R}^m \\ \epsilon\dot{y} &= g(t,x,y,\epsilon), \qquad y \in \mathbb{R}^n \end{aligned} \tag{3.1}$$

where ϵ is a positive scalar parameter multiplying some of the states. When ϵ is set to zero, it causes a fundamental and abrupt change in the dynamic properties of the system. Instead of having a set of equations which are purely differential, we are left with the set of combined differential and algebraic (or transcendental) equations

$$\begin{aligned} \dot{x} &= f(t,x,y,0) \\ 0 &= g(t,x,y,0). \end{aligned} \tag{3.2}$$

The system of equation (3.2) is a lower order system than the one described by equation (3.1) and in many cases would be easier to analyze: a motivation for performing this operation. However, the two systems may be very different because of the discontinuity introduced by setting ϵ to zero. Singular perturbation theory surmounts that problem by analyzing the standard model in different time-scales. Informally stated, the differing time-scales argument considers the *reduced system* given by equation (3.2) to be the slow response of the full system given by equation (3.1). The discrepancy between the response of the full model and the reduced model is the fast transient and is modeled by the *boundary-layer system.* The assumption is that the reduced system approaches the original full system as they evolve beyond the initial boundary-layer interval. The formal exposition of these ideas requires the introduction of the concept of the *integral manifold.*

Definition 3.1 (Integral manifold)
A smooth manifold $S \subset \mathbb{R} \times \mathbb{R}^n$ is called an *integral manifold* for the equation

$$\dot{x} = X(t,x), \qquad x \in \mathbb{R}^n$$

if for all $(t_0, x_0) \in S$, the solution $(t, x(t)) \in S, x(t_0) = x_0$, for $t \in \mathbb{R}$. If this is true only for a restricted t then S is a local integral manifold for the equation.

We will use the terms integral manifold and *invariant manifold* interchangeably in the sequel. The following theorem sets up the necessary conditions for the existence of an integral manifold for the standard model [51].

Theorem 3.1 (Existence of an integral manifold for the standard model)
Let the system of equations

$$\dot{x} = f(t,x,y,\epsilon) \qquad x \in \mathbb{R}^m, \tag{3.3}$$

$$\epsilon\dot{y} = g(t,x,y,\epsilon) \qquad y \in \mathbb{R}^n \tag{3.4}$$

where $t \in \mathbb{R}$ and ϵ is a small positive parameter satisfy the following conditions:

1. *$g(t,x,y,0) = 0$ has an isolated solution $y_0 = h_0(t,x)$ for $t \in R, x \in \mathbb{R}^m$.*
2. *The functions f, g and h_0 are twice continuously differentiable in t and x, $t \in \mathbb{R}$, $x \in \mathbb{R}^m$, $|y - h_0(t,x)| \leqslant \rho$, $0 \leqslant \epsilon \leqslant \epsilon_0$, for some $\rho, \epsilon_0 > 0$.*

3. *The eigenvalues* $\lambda_i = \lambda_i(t,x), i = 1,2,\ldots,n$, *of the matrix*

$$B(t,x) = \frac{\partial g}{\partial y}(t,x,h_0(t,x),0)$$

satisfy the inequality

$$\mathrm{Re}(\lambda_i) \leqslant -2\beta < 0, \quad t \in \mathbb{R}, x \in \mathbb{R}^m,$$

for some $\beta > 0$.

Under these assumptions the system of equations (3.3, 3.4) has the integral manifold $y_\epsilon = h(t,x,\epsilon)$ *on which the flow of the system is governed by the m-dimensional equations* $\dot{x} = f(t,x,h(t,x,\epsilon),\epsilon)$. *The function h is continuously differentiable and* $h(t,x,0) = h_0$.

Using the definition of the integral manifold $y_\epsilon = h(t,x,\epsilon)$, we can get a condition to verify if the manifold given by $y = h(t,x)$ is actually an integral manifold:

$$\begin{aligned} \epsilon\dot{y} &= \epsilon\dot{h} = \epsilon\frac{\partial h}{\partial t} + \epsilon\frac{\partial h}{\partial x}\dot{x}, \\ &= \epsilon\frac{\partial h}{\partial t} + \epsilon\frac{\partial h}{\partial x}f(t,x,h,\epsilon), \quad \text{(on the integral manifold)}. \end{aligned}$$

Therefore for the manifold to be an integral manifold it must satisfy

$$\epsilon\frac{\partial h}{\partial t} + \epsilon\frac{\partial h}{\partial x}f(t,x,h,\epsilon) = g(t,x,h,\epsilon).$$

When f and g are sufficiently smooth then by algebraic operations on the above relation we can also find the asymptotic expansion $h = h_0(t,x) + \epsilon h_1(t,x) + \epsilon^2 \ldots$ for h (for details refer to [23]).

The existence of an integral manifold for a system implies that there is a *reduced order system* which evolves on the manifold and is governed by

$$\dot{x} = f(t,x,h(t,x,\epsilon),\epsilon). \tag{3.5}$$

Thus the integral manifold provides a way of model simplification. However, the reduced order model is a correct description of the full dynamic system only when the initial state is on the reduced manifold. When the initial state is not on the integral manifold we can still use to advantage the concept of the integral manifold by making a change of coordinates from y to the so called *off-manifold* coordinates z,

$$z = y - h(t,x,\epsilon).$$

The description of the system in these coordinates leads to the simple manifold condition

$$z = 0,$$

and on the manifold surface the "z-subsystem" is at an equilibrium in the sense that $z(t_0) = 0$ implies $z(t) = 0$ for all t greater than t_0 for which the solution is defined. The off-manifold coordinates in a sense measure how far away the system is from being on the integral manifold. If our primary interest is in the reduced system, the off-manifold variable can be treated as a correction term or a perturbation of the "main system" represented by equation (3.5).

3.1.2 Time-scale properties of the standard model

In the singular perturbation approach the parameter ϵ which appears in our statement of Theorem 3.1 is the perturbation parameter. The structure of the standard model causes a multi-time-scale response of the system characterized by the presence of fast transients and slower long term behavior. As $\epsilon \to 0$, $\dot{y} = \frac{g}{\epsilon} \to \infty$. Thus the y variables change very rapidly compared to the x variables. On the introduction of a *fast-time* variable which is scaled from the regular time variable by the inverse of ϵ, the slow subsystem appears to change very slowly for small ϵ and is in fact stationary when ϵ is 0. If the other (*fast*) subsystem is stable about the integral manifold (which is to say that the off-manifold coordinates go to 0) in the fast time, then the whole system very soon (in terms of the slow time) "hits" the integral manifold and starts evolving on it. Thus if the slow subsystem (which at this point is the only non-stationary subsystem) has a unique solution and is stable the whole system is essentially stable. Note that this argument is for the case when $\epsilon = 0$. If ϵ is not zero but is small then the conditions of Theorem 3.1 are sufficient to ensure that the evolution of the system on the integral manifold defined by $y_\epsilon = h(t, x, \epsilon)$ is only $\mathcal{O}(\epsilon)$ away from that on the integral manifold h_0 obtained with $\epsilon = 0$.

We state the foregoing ideas mathematically now. In what follows we assume $0 \leqslant \epsilon \leqslant \epsilon_0$, as in Theorem 3.1, and the subscript $_*$ denotes variables on the reduced manifold with $\epsilon = 0$. Note that the specified initial condition $y(t_0)$ can be arbitrarily far from $y_*(t_0) = h(t_0, x_*(t_0))$. Thus in a time interval of interest $[t_0, T]$, $T > t_0$, which is sufficiently long we can have at best

$$y = y_*(t) + \mathcal{O}(\epsilon), \quad t \in [t_1, T], \quad t_1 > t_0, \tag{3.6}$$

with $T > t_1$. However x can start from its specified initial value $x(t_0)$ and

$$x = x_*(t) + \mathcal{O}(\epsilon), \quad t \in [t_0, T]. \tag{3.7}$$

The time interval $[t_0, t_1]$ is the boundary-layer interval. during this time interval y approaches y_* and from t_1 on remains close to it. Defining the fast time variable

$$\mathfrak{s} = \frac{t - t_0}{\epsilon} \quad \Longrightarrow \quad \epsilon \frac{dy}{dt} = \frac{dy}{d\mathfrak{s}},$$

the equation for the fast, boundary-layer subsystem in the fast time is obtained

from equation (3.4):

$$\frac{d\hat{y}}{d\mathfrak{s}} = g(t_0, x(t_0), \hat{y}(\mathfrak{s}), 0), \tag{3.8}$$

with initial condition $\hat{y}(0) = y(t_0)$ and $x(t_0), t_0$ as fixed parameters. A uniform approximation of y can now be given as

$$y = y_*(t) + \hat{y}(\mathfrak{s}) - y_*(t_0) + \mathcal{O}(\epsilon). \tag{3.9}$$

To satisfy the assumptions of Theorem 3.1 we require that the boundary-layer system be asymptotically stable uniformly in $x(t_0)$ and t_0. The boundary-layer interval $[t_0, t_1]$ can be made arbitrarily small by making ϵ sufficiently small. The following theorem by Hoppensteadt [16] summarises and extends the foregoing discussion.

Theorem 3.2 (Singular perturbations on the infinite interval) *Given sufficiently small initial conditions and $\epsilon > 0$, the solution of the full system, equation (3.1), exists for $t_0 \leq t < \infty$, and this solution converges to the solution of the reduced system given by equation (3.2) as $\epsilon \to 0^+$ uniformly on all closed subsets of $t_0 < t < \infty$ if the following conditions are satisfied:*

1. *The reduced system has solutions $x = x(t)$, $y = y(t)$ which exist for all $t_0 \leq t < \infty$.*
2. *The functions f, g and their first partial derivatives f_x, f_y, g_t, g_x and g_y are continuous.*
3. *The function $y_0 = h_0(t, x)$, is an isolated solution of $g(t, x, y, 0) = 0$ for all $t \in \mathbb{R}$ and $x \in \mathbb{R}^m$ and is bounded and twice continuously differentiable.*
4. *The function f is continuous at $y = 0$, $\epsilon = 0$ uniformly in t and x, and $f(t, x, 0, 0)$ and $f_x(t, x, 0, 0)$ are bounded on $\mathbb{R} \times \mathbb{R}^m$.*
5. *The function g is continuous at $\epsilon = 0$ uniformly in t, x and y and $g(t, x, y, 0)$ and its derivatives with respect to t, x and y are bounded on $\mathbb{R} \times \mathbb{R}^{m+n}$.*
6. *The reduced system given by equation (3.2) is uniformly asymptotically stable.*
7. *The boundary layer system given by equation (3.8) is uniformly asymptotically stable uniformly in $t_0 \in \mathbb{R}$ and $x(t_0) \in \mathbb{R}^m$*

The following theorem from Khalil [23] discusses requirements for asymptotic stability:

Theorem 3.3 (Asymptotic stability) *Consider the autonomous singularly perturbed system*

$$\begin{aligned} \dot{x} &= f(x, y), \qquad x \in \mathbb{R}^m \\ \epsilon\dot{y} &= g(x, y), \qquad y \in \mathbb{R}^n. \end{aligned} \tag{3.10}$$

and assume that the origin, $x = 0$, $y = 0$, is an isolated equilibrium for it. Let $y_*(x)$ be y variable on the reduced manifold and $z = y - y_*$ the off-manifold coordinates. The reduced system can be written as

$$\dot{x} = f(x, y_*) \tag{3.11}$$

and the boundary layer system can be written as

$$\frac{dz}{d\tau} = g(x, z + y_*). \tag{3.12}$$

Assume there exists a Lyapunov function $W(x, z)$ for the boundary layer system such that its origin is asymptotically stable uniformly in x and a Lyapunov function $V(x)$ for the reduced system such that its origin is asymptotically stable. Let the following conditions be satisfied for $\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma \geq 0$:

1. $\dot{V} \leq -\alpha_1 \rho_1^2(x)$, *where $\rho_1 : \mathbb{R}^m \to \mathbb{R}$ is a positive definite function.*
2. $W' \leq -\alpha_2 \rho_2^2(z)$, *where $\rho_2 : \mathbb{R}^n \to \mathbb{R}$ is a positive definite function.*
3. $\zeta_1(\|z\|) \leq W(x, z) \leq \zeta_2(\|z\|)$ *for ζ_1 and ζ_2 strictly increasing functions with $\zeta_1(0) = \zeta_2(0) = 0$.*
4. $\frac{\partial V}{\partial x}\big(f(x, z + y_*) - f(x, y_*)\big) \leq \beta_1 \rho_1(x) \rho_2(z)$.
5. $\left(\frac{\partial W}{\partial x} - \frac{\partial W}{\partial y}\frac{\partial y_*}{\partial x}\right) f(x, z + y_*) \leq \beta_2 \rho_1(x) \rho_2(z) + \gamma \rho_2^2(z)$.

Then for

$$\epsilon^* = \frac{\alpha_1 \alpha_2}{\alpha_1 \gamma + \beta_1 \beta_2}$$

the origin of the autonomous singularly perturbed system, equation 3.10, is asymptotically stable for $0 < \epsilon < \epsilon^*$.

The foregoing discussion sets up our use of singular perturbation as a tool for analysis and control. Of course, the model has to be first cast into the framework of the standard model, which means a perturbation parameter has to be chosen. This is nontrivial task in many cases. Once in the framework, if the fast, boundary-layer dynamics are asymptotically stable we can restrict our attention to the reduced system to determine stability or other system properties.

3.1.3 Singular perturbation of second-order ODEs

The standard singular perturbation model is set up in the framework of first-order ODE's. However, dynamic equations for robot manipulators are usually derived in the form of second-order equations, and are intuitively more attractive to manipulate in this form. This section addresses the problem of modifying the approach presented above to handle second-order equations.

One possible way of dealing with second-order equations which cannot be expressed directly in the standard form (equation (3.1)) is that of Hoppensteadt in [17]. The problem addressed is

$$\begin{aligned} \dot{x} &= f(t, x, y_1, \ldots, y_n, \epsilon), \\ \epsilon_j \dot{y}_j &= g_j(t, x, y_1, \ldots, y_n, \epsilon), \quad j = 1, \ldots, n, \end{aligned}$$

with, x, y_j and f, g_j respectively $\in \mathbb{R}^n, \mathbb{R}^{n_j}$, $|\epsilon| \to 0$, $\epsilon_j > 0$, $\epsilon_{j+1}/\epsilon_j \to 0$ as $|\epsilon| \to 0$. Conditions for the solutions of the above equation to converge to those of

$$\begin{aligned} \dot{x} &= f(t, x, y_1, \ldots, y_n, 0), \\ 0 &= g_j(t, x, y_1, \ldots, y_n, 0), \quad j = 1, \ldots, n, \end{aligned}$$

require the consideration of a hierarchy of boundary-layer equations. However, this approach need not be used in our problem because we are able to frame our problem in a way which can easily be transformed to the standard model.

A set of second-order ODE's can be written as

$$\ddot{q} = c(t, q, \dot{q}), \quad Q \in \mathbb{R}^k.$$

If

$$q = \begin{bmatrix} \Theta \\ \Phi \end{bmatrix},$$

the second order equations can be written as

$$\begin{aligned} \ddot{\Theta} &= f(t, \Theta, \dot{\Theta}, \Phi, \dot{\Phi}), \quad \Theta \in \mathbb{R}^m \\ \ddot{\Phi} &= g(t, \Theta, \dot{\Theta}, \Phi, \dot{\Phi}), \quad \Phi \in \mathbb{R}^n, \end{aligned}$$

with $k = m + n$. On introduction of the perturbation parameter ϵ, assume that the equations can be expressed in terms of Θ and a new variable Ψ as

$$\begin{aligned} \ddot{\Theta} &= \tilde{f}(t, \Theta, \dot{\Theta}, \Psi, \epsilon\dot{\Psi}, \epsilon), \quad \Theta \in \mathbb{R}^m \\ \epsilon^2 \ddot{\Psi} &= \tilde{g}(t, \Theta, \dot{\Theta}, \Psi, \epsilon\dot{\Psi}, \epsilon), \quad \Psi \in \mathbb{R}^n. \end{aligned} \tag{3.13}$$

Using

$$X = \begin{bmatrix} \Theta \\ \dot{\Theta} \end{bmatrix} \quad \text{and,} \quad Y = \begin{bmatrix} \Psi \\ \epsilon\dot{\Psi} \end{bmatrix},$$

we can write equation (3.13) as

$$\begin{aligned} \dot{X} &= \tilde{f}(t, X, Y, \epsilon), \quad X \in \mathbb{R}^{2m}, \\ \epsilon\dot{Y} &= \tilde{g}(t, X, Y, \epsilon), \quad Y \in \mathbb{R}^{2n}, \end{aligned} \tag{3.14}$$

which is in the form of the standard model, equation (3.1).

Obviously, this approach works only if we can find the correct perturation parameter ϵ to convert the second-order equations to the rather special form required.

3.2 Treating Flexibility Using Singular Perturbation

Mechanical systems with flexibilities have been analyzed using tools from singular perturbation analysis for a long time. Analysis of flexibility using singular perturbation is based on the assumption that the system modes can be separated into two distinct groups; low frequency modes, which can be considered the slow modes, and high frequency modes which are fast modes.

3.2.1 Traditional singular perturbation approach for flexibility

Traditionally, the perturbation parameters (ϵ), used for the analysis of flexibility are typically the inverse of the stiffness of the flexible mechanism or the inverse of the stiffness weighted by a factor depending on the mass (see for example [22]). This framework results in the reduced system being rigid. As the full system is only a perturbation away from the reduced system, this approach can handle relatively little flexibility.

In the case of robot manipulators, singular perturbation techniques have been used previously to deal with joint flexibility [52]. The problem of joint flexibility is significantly easier than that of link flexibility due to the localized nature of joint flexibility. In the method presented in [52] the dynamic model of the robot is extended with the inclusion of an additional configuration variable at each joint to allow the actuator and the link angle to be different due to the flexibility. Torsional springs at the joints are used to model the flexibility. The reduced system is obtained by setting the torsional spring constant to be infinite (the perturbation parameter is the inverse of the spring constant). The flexible joint robot is therefore a perturbation of the rigid robot.

In the literature there is also discussion of perturbation techniques (both regular and singular) for flexible link manipulators [9]. The starting point for the singular perturbation approach is a dynamic model of the arm. As the ultimate aim in these analyses is to separate the rigid and the flexible dynamics, the modeling is done so as to be able to distinguish between the two effects. Such an analysis is carried out in [49, 50], and controllers are designed for the rigid and the flexible subsystems. This model is usually written as [50]

$$M(\Theta,\Psi)\begin{bmatrix}\ddot{\Theta}\\ \ddot{\Psi}\end{bmatrix}+\begin{bmatrix}C_1(\Theta,\dot{\Theta},\Psi,\dot{\Psi})\\ C_2(\Theta,\dot{\Theta},\Psi,\dot{\Psi})\end{bmatrix}+\begin{bmatrix}0\\ K\Psi\end{bmatrix}=\begin{bmatrix}\tau\\ 0\end{bmatrix},$$

where, the Θ are the joint angles and the Ψ are the variables introduced for modeling flexibility. The $K\Psi$ term is a measure of the flexibility. The 0 on the right hand side of the equation denotes that forces/torques may not be applied directly to the flexible variables. To carry out the singular perturbation argument the assumption is made that the dynamics of the arm can be partitioned into the fast dynamics,

which are due to the flexibility and the slow dynamics, which are due to the rigid-body modes. To enforce the high frequency dynamics of the flexibility, the stiffness of the manipulator has to be high (high K). Thus, the perturbation parameters used in these analyses have always been scaled from the inverse of the stiffness associated with the manipulator. The reduced system ($\epsilon = 0, \text{or}, K \rightarrow \infty$) is rigid. Small values of the perturbation parameter correspond to systems which have a "small" amount of flexibility. Thus these analyses present results useful for systems which are "close to" being rigid. Even so it is doubtful that the assumption of separation is justified, particularly for high speed motion [9]. The assumption is even more questionable when we consider constrained motion of the manipulator. Due to the constraint, any motion of the "rigid variables" has to be matched by the motion of the "flexible variables" so as to maintain the constraint. Thus, the separation of the frequencies does not seem to hold.

3.2.2 Singular perturbation approach for treating relatively large flexibility

Given that our stated aim is to model relatively large flexibility in constrained motion robotic tasks, the traditional approach to singular perturbation modeling and analysis of flexible systems is clearly inadequate. In this section we present our alternative approach to singular perturbation analysis of flexible robots which is capable of handling significant flexibility.

Before embarking on a description of our approach we present the following as a motivation and an illustration of the ideas which form its basis. Consider a scalar, linear, mechanical system

$$m\ddot{x} + c\dot{x} + kx = f(t).$$

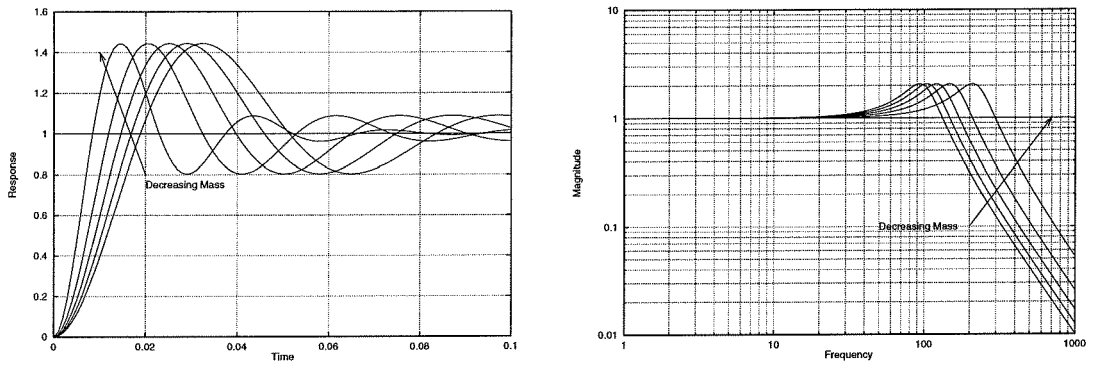
The *damping factor* (or damping ratio) of the system is given by

$$\zeta = \frac{c}{2\sqrt{km}}. \tag{3.15}$$

The value of ζ is very important in determining the transient response of a system. If $\zeta = 1$ the system is critically damped, $\zeta > 1$ is an over-damped system and $\zeta < 1$ is an under-damped system. Critically and over-damped systems do not exhibit oscillatory behavior. We assume that the transient response of the flexible beam is its most important characteristic and therefore must be preserved in any analysis. To conserve the value of ζ irrespective of m, the damping coefficient c must be of the form

$$c = b\sqrt{m}, \tag{3.16}$$

where b is a constant. Figure 3.1 shows the behavior of this system as the mass is varied. The step response becomes faster with decreasing mass. However, the overshoot of the system remains a constant. The Bode plot shows that with decrease in mass the natural frequency of the system increases, however the maximum response



(a) Step response.

(b) Bode plot.

Figure 3.1 Spring–mass–damper system with varying mass and constant damping ratio.

does not change. Note that k, which models the flexibility, has been kept a constant. In this system if the mass of the system $\to 0$ the system responds infinitely fast to any input, however, the extent of its response is not affected. This behavior is the rationale behind the singular perturbation setup we use for analyzing flexible manipulators.

Our approach for singular perturbation of flexible link robots does not use the inverse of the flexibility as the perturbation parameter. Instead we choose the mass of the flexible sublinks as the square of the perturbation parameter, that is,

$$\epsilon^2 = m_{fl}, \tag{3.17}$$

where, m_{fl} is the mass of the flexible sublinks. All sublinks do not have to have the same mass. Their masses must however be scaled by ϵ^2. As we want to preserve the quality of the transient response of the system, we make the additional assumption that the damping ratio (or factor) for the flexible links is a constant irrespective of the change in mass. Therefore the damping at the unactuated joints in the sublink model of the manipulator, k_{f_2} in equation (2.42), is scaled by the perturbation parameter (from equation (3.16)). Hence as the perturbation parameter ϵ goes to 0 we have a simultaneous reduction in the mass and the damping. The extent of the response is however not affected.

Comments on the new approach

The singular perturbation approach described above is different from the usual singular perturbation that has been used so far for the treatment of flexible mechanical systems. This approach offers significant advantages for the analysis of flexible robots. The most important advantage we gain is the ability to consider

significant flexibility. We do not have to assume that the reduced system is a rigid robot, indeed the reduced robot has exactly the same flexibility as the original flexible robot, and flexes to the same extent. For hybrid force/position tasks this is very important because the forces to be applied at the end effector depend greatly on the geometry of the system. The preservation of the damping factor preserves the qualitative transient behavior of the manipulator. We shall discuss later the dynamic properties of the reduced, singularly perturbed system and its behavior.

Another feature of this approach is its suitability for application to real systems. In Chapter 1 it was pointed out that one of the primary reasons for wanting to design and operate flexible robots is to benefit from the reduction in mass. An analysis based on our approach should work very well for lightweight robots. The linking together of the variation in mass and the damping of the flexibility results in a system for which we can analytically prove the existence of stable control laws. Thus it provides guidelines for the design of light flexible manipulators. Control laws designed for flexible robots under our singular perturbation framework will perform qualitatively similarly for classes of flexible robots with different masses, if the damping present in the flexibility is varied as in our approach. The parameters available during design include choice of material, geometry and actuation methods. Materials differ in their densities and viscoelastic properties. The flexibility of links are also affected greatly by the geometric distribution of the material. Actuation, especially with the advent of smart materials like shape-memory alloys can be used to change properties of links even in real-time and with feedback. Given these choices during the design stage, it is possible to preserve, the relation between the mass and the viscous damping of flexible links.

3.3 Singular Perturbation Based Reduction of the Flexible Manipulator System

Similar to equation (2.42), we can write the dynamic equation of a constrained flexible robot (compare equation (2.22)) as

$$\begin{bmatrix} m_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} \ddot{\Theta} \\ \ddot{\Psi} \end{bmatrix} + \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & k_{s\psi} \end{bmatrix} \begin{bmatrix} \Theta \\ \Psi \end{bmatrix} + \begin{bmatrix} k_{f\theta} & 0 \\ 0 & K_{f\psi} \end{bmatrix} \begin{bmatrix} \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} + \begin{bmatrix} A_\theta \\ A_\psi \end{bmatrix} \lambda = \begin{bmatrix} \tau \\ 0 \end{bmatrix}. \quad (3.18)$$

As in equation (2.42), we have used Θ to represent the actual angular variables of the robot and Ψ to represent the passive joint angles of the sublink model. The constraint condition is

$$h(\Theta, \Psi) = 0, \quad (3.19)$$

and

$$A_\theta = \frac{\partial h}{\partial \Theta}^T \qquad A_\psi = \frac{\partial h}{\partial \Psi}^T . \tag{3.20}$$

The spring constants modeling the flexibility are the diagonal submatrix k_{sf} and the viscous damping of the active and passive joints are the diagonal submatrices $k_{f\theta}$ and $K_{f\psi}$ respectively. We have written the dynamic equations in the above form to facilitate the application of our singular perturbation approach. The significance of uppercase and lowercase symbols will be apparent in the sequel, when the perturbation parameter is introduced. The structural properties of the robot dynamic equations, set out in Lemma 2.1 are valid for the quantities in this equation. Hence,

$$\dot{M} - 2C = \begin{bmatrix} \dot{m}_{11} & \dot{M}_{12} \\ \dot{M}_{21} & \dot{M}_{22} \end{bmatrix} - 2 \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \tag{3.21}$$

is skew symmetric. Therefore, the diagonal submatrix blocks $\dot{m}_{11} - 2C_{11}$ and $\dot{M}_{22} - 2C_{22}$ must also be skew-symmetric.

We assume that the masses of all the flexible links are scaled by the parameter ϵ^2 and the viscous damping at the passive joints is scaled by ϵ, i.e.,

$$K_{f\psi} = \epsilon k_{f\psi}. \tag{3.22}$$

Using the definition of the Coriolis matrix (equation (2.9)) and substituting in the assumptions made above we can rewrite the dynamic equations of the flexible manipulator as

$$\begin{bmatrix} m_{11} & \epsilon^2 m_{12} \\ \epsilon^2 m_{21} & \epsilon^2 m_{22} \end{bmatrix} \begin{bmatrix} \ddot{\Theta} \\ \ddot{\Psi} \end{bmatrix} + \begin{bmatrix} c_{11}\dot{\Theta} & \epsilon^2 c_{12}\dot{\Theta} \\ \epsilon^2 c_{21}\dot{\Psi} & \epsilon^2 c_{22}\dot{\Psi} \end{bmatrix} \begin{bmatrix} \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & k_{s\psi} \end{bmatrix} \begin{bmatrix} \Theta \\ \Psi \end{bmatrix} + \begin{bmatrix} k_{f\theta} & 0 \\ 0 & \epsilon k_{f\psi} \end{bmatrix} \begin{bmatrix} \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} + \begin{bmatrix} A_\theta \\ A_\psi \end{bmatrix} \lambda = \begin{bmatrix} \tau \\ 0 \end{bmatrix}, \tag{3.23}$$

where we have used

$$\begin{aligned} M_{ji} &= \epsilon^2 m_{ij} \quad \text{(except for } ij = 11\text{)} \\ C_{11} &= c_{11}\dot{\Theta} \\ C_{12} &= \epsilon^2 c_{12}\dot{\Theta} \\ C_{ij} &= \epsilon^2 c_{ij}\dot{\Psi} \quad \text{(for } i = 2, j = 1, 2\text{) and} \\ K_{f\psi} &= \epsilon k_{f\psi}. \end{aligned}$$

Equations (3.18) and (3.23) are identical equations. Transforming the above to the form of equation (3.13) we get

$$\begin{bmatrix} m_{11} & m_{12} \\ \epsilon^2 m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} \ddot{\Theta} \\ \epsilon^2 \ddot{\Psi} \end{bmatrix} = \begin{bmatrix} \tau \\ 0 \end{bmatrix} - \left(\begin{bmatrix} c_{11}\dot{\Theta} & \epsilon^2 c_{12}\dot{\Theta} \\ \epsilon^2 c_{21}\dot{\Psi} & \epsilon^2 c_{22}\dot{\Psi} \end{bmatrix} \begin{bmatrix} \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & k_{s\psi} \end{bmatrix} \begin{bmatrix} \Theta \\ \Psi \end{bmatrix} + \begin{bmatrix} k_{f\theta} & 0 \\ 0 & \epsilon k_{f\psi} \end{bmatrix} \begin{bmatrix} \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} + \begin{bmatrix} A_\theta \\ A_\psi \end{bmatrix} \lambda \right), \tag{3.24}$$

or,

$$\begin{bmatrix} \ddot{\Theta} \\ \epsilon^2\ddot{\Psi} \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} \\ \epsilon^2 m_{21} & m_{22} \end{bmatrix}^{-1} \left(\begin{bmatrix} \tau \\ 0 \end{bmatrix} - \begin{bmatrix} c_{11}\dot{\Theta} & \epsilon^2 c_{12}\dot{\Theta} \\ \epsilon^2 c_{21}\dot{\Psi} & \epsilon^2 c_{22}\dot{\Psi} \end{bmatrix} \begin{bmatrix} \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & k_{s\psi} \end{bmatrix} \begin{bmatrix} \Theta \\ \Psi \end{bmatrix} - \begin{bmatrix} k_{f\theta} & 0 \\ 0 & \epsilon k_{f\psi} \end{bmatrix} \begin{bmatrix} \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} - \begin{bmatrix} A_\theta \\ A_\psi \end{bmatrix} \lambda \right). \quad (3.25)$$

Equation (3.25) is the dynamic equations of the flexible manipulator transformed to the standard form for second-order ODE's described in equation (3.13). We will not convert it further to the first-order standard form of equation (3.14), but will manipulate it in its second-order form. In equation (3.25) the singular perturbation parameter is ϵ, the fast variables are the Ψ and the variables governing the evolution on the reduced manifold are the Θ.

3.3.1 Reduced order system

The reduced order system is obtained by setting $\epsilon = 0$ in equation (3.25):

$$m_{11}\ddot{\Theta} + C_{11}\dot{\Theta} + k_{f\theta}\dot{\Theta} + A_\theta\lambda = \tau \quad (3.26)$$

$$k_{s\psi}\Psi + A_\psi\lambda = 0 \quad (3.27)$$

$$h(\Theta, \Psi) = 0. \quad (3.28)$$

Consider the equations (3.27) and (3.28). Applying the implicit function theorem to these equations we can say that if the matrix

$$\Omega = \begin{bmatrix} k_{s\psi} + \frac{\partial A_\psi \lambda}{\partial \Psi} & A_\psi \\ A_\psi^T & 0 \end{bmatrix}$$

is nonsingular then we can express Ψ and λ as functions of Θ. This is assured at all nonsingular configurations of the flexible links of the manipulator by the presence of A_ψ at the anti-diagonal positions of Ω. Therefore, in the reduced system we can use

$$\Psi = \Psi(\Theta) \qquad \text{and}$$
$$\lambda = \lambda(\Theta). \quad (3.29)$$

The reduced order system is the system for which we shall design some of our control laws. Stability of the laws will be proven for the reduced order system. The theory of singular perturbations presented earlier will be then be used to draw conclusions about the full system.

3.3.2 Boundary-layer system

We define the fast time variable $s = \frac{t}{\epsilon}$. Differentiation with respect to this time variable is denoted by $'$. The derivatives of the "slow" variables (Θ) in the new

time, $\mathfrak{s}$ are small and disappear entirely at $\epsilon = 0$. In the boundary layer system therefore Θ are stationary. Hence,

$$\Theta' = 0 \quad \text{and,} \quad \Theta'' = 0.$$

Thus we have the following relation which holds for the constraints in the boundary layer system:

$$\begin{aligned} & h(\Theta, \Psi) && = 0, \\ \Longrightarrow \quad & \tfrac{dh}{d\mathfrak{s}} && = 0, \\ \Longrightarrow \quad & \Theta'^T A_\theta + \Psi'^T A_\psi && = 0, \\ \Longrightarrow \quad & \Psi'^T A_\psi && = 0. \end{aligned} \tag{3.30}$$

Setting $\epsilon = 0$ in equation (3.25) and using the following identity for the inverse of a matrix in terms of its block sub-matrices:

$$\begin{bmatrix} A & D \\ 0 & B \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} & -A^{-1}DB^{-1} \\ 0 & B^{-1} \end{bmatrix},$$

the boundary layer system is

$$\begin{aligned} m_{22}\epsilon^2\ddot{\Psi} + (\epsilon c_{22}\dot{\Psi})\epsilon\dot{\Psi} + k_{f_\psi}\epsilon\dot{\Psi} + k_{s\psi}\Psi + A_\psi\lambda = 0, \\ h(\Theta, \Psi) = 0. \end{aligned} \tag{3.31}$$

In the fast time variable this can be written as

$$\begin{aligned} m_{22}\Psi'' + \tilde{c}_{22}\Psi' + k_{f_\psi}\Psi' + k_{s\psi}\Psi + A_\psi\lambda = 0, \\ \Psi'^T A_\psi = 0, \end{aligned} \tag{3.32}$$

where we have used

$$\begin{aligned} \epsilon\frac{d}{dt} &= \frac{d}{d\mathfrak{s}}, \\ \epsilon^2\frac{d^2}{dt^2} &= \frac{d^2}{d\mathfrak{s}^2}, \qquad \text{and} \\ \tilde{c}_{22} &= c_{22}\epsilon\dot{\Psi}. \end{aligned}$$

Theorem 3.4 (Equilibrium of the boundary-layer system) *Assuming $k_{f\psi} > 0$, the system defined by the dynamic equations* (3.32) *converges asymptotically to*

$$k_{s\psi}\Psi + A_\psi\lambda = 0,$$

and at equilibrium is restricted to it.

Proof: To prove stability of the boundary layer system we use the candidate Lyapunov function

$$V = \frac{1}{2}\Psi'^T m_{22}\Psi' + \frac{1}{2}\Psi^T k_{s\psi}\Psi.$$

As discussed previously (equation (3.21)) $\dot{M}_{22} - 2C_{22}$ is skew-symmetric. Seeing that

$$\begin{aligned}\dot{M}_{22} - 2C_{22} &= \epsilon^2 \dot{m}_{22} - 2\epsilon^2 c_{22}\dot{\Psi} \\ &= \epsilon m'_{22} - 2\epsilon c_{22}\epsilon\dot{\Psi} \\ &= \epsilon(m'_{22} - 2\tilde{c}_{22}),\end{aligned}$$

we can conclude $m'_{22} - 2\tilde{c}_{22}$ is also skew-symmetric.

The time derivative of the candidate Lyapunov function in the fast time is

$$\begin{aligned}V' &= \Psi'^T(\frac{1}{2}m'_{22}\Psi' - \tilde{c}_{22}\Psi' - k_{f_\psi}\Psi' - k_{s\psi}\Psi - A_\psi\lambda + k_{s\psi}\Psi) \\ &= \Psi'^T(-k_{f_\psi}\Psi' - k_{s\psi}\Psi - A_\psi\lambda + k_{s\psi}\Psi), \qquad \text{(skew-symmetry of } m'_{22} - 2\tilde{c}_{22}) \\ &= -\Psi'^T k_{f_\psi}\Psi'. \qquad \text{(using equation (3.30))}\end{aligned}$$

Assuming positive definiteness of k_{f_ψ}, the boundary-layer subsystem stops moving in the fast time scale, i.e.

$$\Psi' = 0 \quad \text{and,} \quad \Psi'' = 0.$$

Substituting the above in equation (3.32) we find that on cessation of motion in the fast time the dynamics are restricted to the submanifold given by

$$k_{s\psi}\Psi + A_\psi\lambda = 0.$$

As we have seen, this is exactly the manifold on which the reduced order subsystem evolves. □

3.3.3 Comments on the two subsystems

The division into the fast and slow subsystems shows some interesting properties of the flexible manipulator system. In our singular perturbation approach the dynamic effects of the flexibility are only seen during the operation of the boundary layer system. This system is not affected by the control torques at all. It is however stable by itself. Indeed, we can use skewing terms in the Lyapunov function to prove its exponential stabilization to the reduced manifold. This is because of the particular way in which the damping of the system is introduced into the system. That elastic material do exhibit viscous damping is well documented. Modern treatments of flexibility model flexible elements in the viscoelastic framework. It is further observed that flexible links under load, for example a flexible link pushing

against a wall, exhibit much higher effective damping than when free; vibrations die out much more rapidly. This fits in favorably with hybrid force/position control applications where links are always under load.

The reduced order system is not a rigid system. It exhibits the full flexibility of the original system. However, the perturbation technique we use results in the flexible dynamics becoming infinitely fast, and therefore they are no longer dynamic. The shape of the beam changes infinitely fast to adjust to the force condition at the tip through the algebraic relation

$$K_s \Psi + A_\psi \lambda = 0.$$

This is different from the beam shape not changing at all, as would be the case for the rigid manipulator. The assumption of the fast change in shape is not a very bad assumption for lightweight links. We shall exploit this property of the reduced system when we design controllers.

Chapter 4

Control of Flexible Link Manipulators

Initial studies of flexible link manipulator control concentrated mainly on unconstrained trajectories [6, 34, 60]. The control task was to follow a specified end-effector trajectory using a robot with flexible links. The primary consideration was the suppression of vibrations induced by the flexible dynamics, thus improving the tracking performance of the robot. More recently the problem of hybrid force/position control using flexible link manipulators has been considered in [27, 32, 33].

In this chapter we propose controllers for the control of significantly flexible robotic manipulators in hybrid force/position control tasks. Our model, outlined in the previous chapters, allows the consideration of significant flexibility. We propose workspace controllers based on the full dynamic model and on the singular perturbation model respectively.

4.1 Problem Setup

Our final aim in developing controllers for flexible robots is to be able to implement grasping with multiple flexible fingers. The setup we aim to control (and which resembles our experimental setup) is shown in Figure 4.1. Treating the full flexible grasping problem is very complex due to the complicated interaction between the multiple subsystems involved. However, as shown in Chapter 2, the modeling and the dynamics associated with the grasping problem are very similar to those of individual constrained robots. The grasping constraints (equation (2.29)) enter the dynamic equations of the grasping system very similarly to workspace constraints for a single robot (equation (2.20)). Thus, analysis carried out with individual robots in constrained workspaces will extend to the grasping situation. Indeed, for each individual finger involved in the grasping task, manipulation of the grasped object appears little different from applying forces against workspace constraints, other than the dynamics of the object being grasped. In the above discussion we are considering stable, grasping without finger rolling only. The argument will not extend to systems undergoing regrasping maneuvers or finger rolling.

Consequent to the above discussion, we consider for our analysis the simpler system shown in Figure 4.2: a single manipulator in a constrained motion task

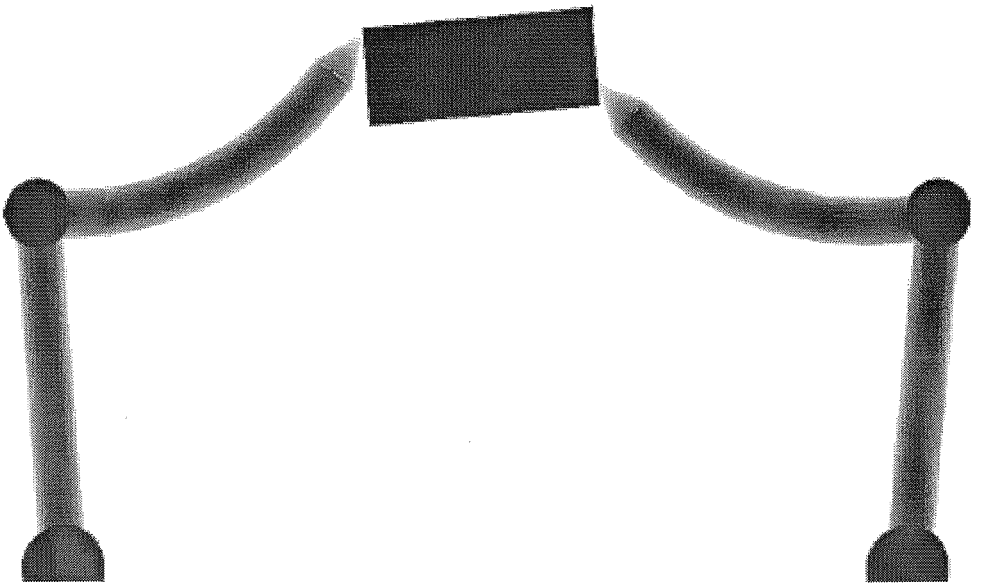


Figure 4.1 Planar grasping setup with last link flexible.

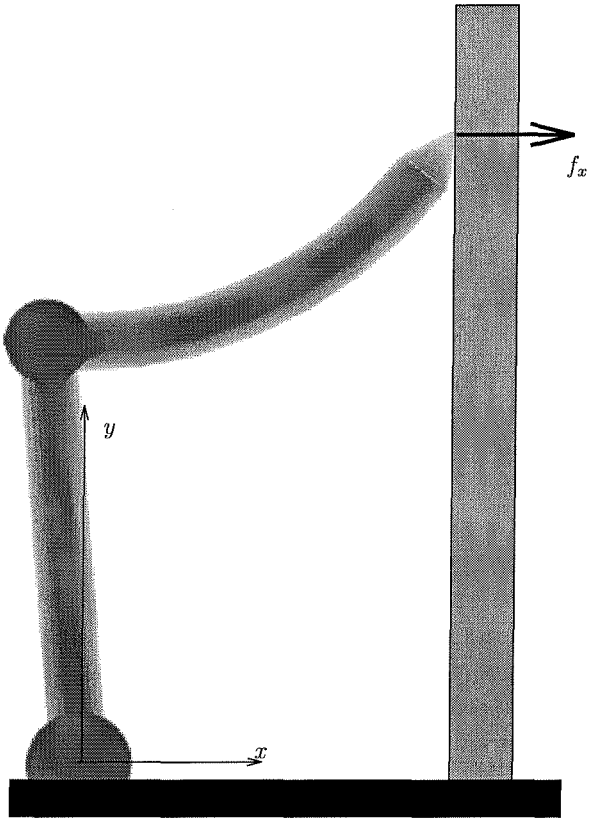


Figure 4.2 Single finger with last link flexible pushing against a wall.

(shown in the figure as pushing against a wall). This results in a simplification of the analysis. Note that though the setup in Figure 4.2 has only the last link flexible resembling our experimental setup, the analysis to follow applies equally to the case of manipulators with all links flexible.

The kinematics of rigid link robots are dependent solely on the joint space variables. The Jacobian transformation (defined in equation (2.16)), derived from the kinematics, relates the workspace and joint space generalized forces (under quasi-static conditions) by

$$\tau = J^T(\Theta)F, \tag{4.1}$$

where we have

- Θ the actuated joint variables,
- τ the actuated joint space generalized forces,
- F the workspace generalized forces, and
- J the workspace Jacobian.

If the current Jacobian is known, the robot actuator forces required to produce the desired workspace forces can be calculated using the above relation. Note that this calculation is true only for the static case. Hence, feedback linearization techniques, like computed torque, are used to compensate for the dynamic forces. All workspace control methods use the workspace Jacobian in some form to convert between workspace and joint space generalized forces.

In the case of robots with flexible links, the kinematics are no longer solely a function of the actuated joint variables. At any configuration of the robot, the application of a workspace force by the robot causes a reaction force on the robot itself, thus changing the "shape" of the robot and therefore its kinematics. The kinematics can therefore no longer be stated independent of the forces acting on the robot. Hence, the application of forces using flexible link robots requires other considerations in addition to those for rigid link robots. For the flexible case we can write

$$\tau = J^T(\Theta, \Psi)F, \tag{4.2}$$

where Ψ are the internal states of the robot associated with its flexibility. Note that in this equation τ represents both the actuated forces as well as the unactuated (flexibility provided) forces. In essence Ψ represents the shape of the flexible links and is dependent not only on the current forces acting on the robot but also on its current configuration. Therefore, for the equilibrium solution

$$\Psi = \Psi(\Theta, \tau, F).$$

If we use the full nonlinear beam model for the flexible link, Ψ is the solution of a system of partial differential equations and is therefore infinite dimensional. The usual modeling methods, described in Chapter 2, reduce the system to a finite

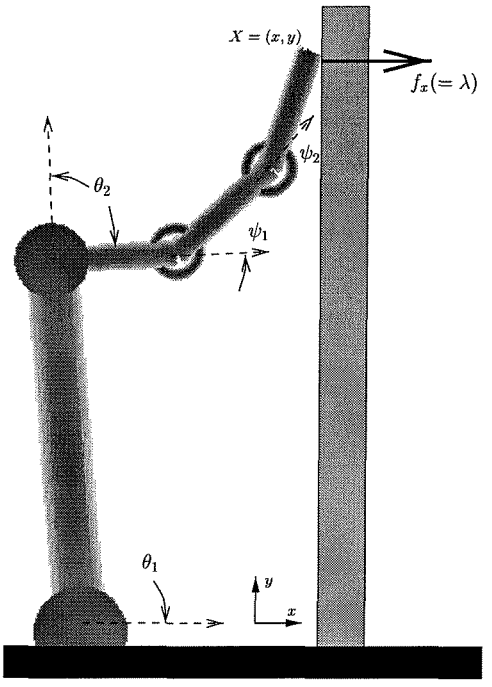


Figure 4.3 Single finger pushing against a wall: sublink model.

dimensional system. However, there still remains the problem of finding the Ψ. The Θ are usually sensed by angle encoders at the joints of the robot. Solutions of the flexibility modeling equations are complex and even assuming that unique, closed form solutions exist (which is not true in general) would be hard to compute in real-time. The other solution would be to have sensors to measure the state of the flexible beam. Such sensors do not yet exist.

For our analysis we use the sublink model of the flexibility described in section 2.3. Figure 4.3 shows the system analyzed. The control objective is to regulate the end-position of the manipulator to the desired point (X_0) and to apply a desired force (λ_0) against the constraint at the desired point. Both the goals are prescribed in the workspace and must be consistent with the constraints on the system. In what follows we use K_p to represent a positive definite, constant, diagonal matrix of proportional gains and K_d to represent a positive definite, constant, diagonal matrix of derivative gains. We also use the subscript 0 to denote quantities at the final equilibrium state. We use the vector

$$q = \begin{bmatrix} \Theta \\ \Psi \end{bmatrix}$$

as the vector of all joint angles. The dynamic equations of the constrained flexible

robot are therefore given by (see equation (2.22))

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + K_s q + K_f \dot{q} + A\lambda = \begin{bmatrix} \tau \\ 0 \end{bmatrix}$$
$$h(q) = 0. \tag{4.3}$$

We show that under certain conditions, it is not required that we know the full state of the flexible robot to be able to control it. We also address issues of applicability of the control laws developed to real systems. Experimental and simulation results for a finger pushing against a wall are presented in this chapter. In later chapters we provide much more detailed simulation results as well as results obtained from using our control laws on an experimental grasping setup.

4.2 Joint PD Controller

The *joint PD controller* applies control actuation based on the values of the actuated joint angles of the flexible robot. This strategy requires no feedback of the tip position of the finger. However, it does require precomputed values of the actuated joint angles and torques at the final equilibrium position desired. It should be pointed out that the computation of the equilibrium torques and angles is relatively simple as it is the solution to a problem of static equilibrium. Even so the procedure for solution is recursive.

The controller action is PD control on the joint angles with an additional feed forward torque on each actuated joint equal to the precomputed equilibrium torque. In joint space the control function is given by

$$\begin{bmatrix} \tau \\ 0 \end{bmatrix} = K_p(q_0 - q) - K_d\dot{q} + \begin{bmatrix} \tau_0 \\ 0 \end{bmatrix}, \tag{4.4}$$

where K_p and K_d are diagonal with the bottom two rows of each consisting entirely of zeroes and q_0 is the desired equilibrium position in joint space. Note that torques cannot be applied to the passive joints. The equilibrium torque required to hold the manipulator at its equilibrium position is τ_0. Before proving that this controller performs the required task we need to state and prove the following lemma.

Lemma 4.1 (Generalized coordinates for constrained manipulator) *The equations for the constrained robotic manipulator*

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + K_s q + K_f \dot{q} + A\lambda = \begin{bmatrix} \tau \\ 0 \end{bmatrix},$$
$$h(q) = 0,$$

$q \in \mathbb{R}^n$, $h(q) : \mathbb{R}^n \to \mathbb{R}^k$, *can be written equivalently in generalized coordinates* $\alpha \in \mathbb{R}^{n-k}$ *as*

$$\tilde{M}\ddot{\alpha} + \tilde{C}\dot{\alpha} + \tilde{K}_s f(\alpha) + \tilde{K}_f \dot{\alpha} = \tilde{\tau},$$

where, $\tilde{M}$ is the mass matrix, $\tilde{C}$ is the Coriolis matrix, $\tilde{K}_s$ is the matrix of spring constants, $\tilde{K}_f$ is the damping matrix and $\tilde{\tau}$ is the vector of joint torques in the new (α) coordinates, obtained by applying the proper coordinate transformations to these quantities in the old (q) coordinates.

Proof: We give a constructive proof of the above. Given k smooth, independent, holonomic constraints $h(q) = 0$, $h(q) : \mathbb{R}^n \rightarrow \mathbb{R}^k$ on a dynamical system with configuration variables $q \in \mathbb{R}^n$ it is always possible to find generalized coordinates α such that $\alpha \in \mathbb{R}^{n-k}$. Further, it is easy to see that $0 \in \mathbb{R}^k$ will be a regular value of the map $h(q)$, i.e., q satisfying $h(q) = 0$ lie on a smooth submanifold of dimension $n - k$. From this we can conclude that there must exist a mapping $f(\alpha) = q$ which when its range is restricted to the submanifold satisfying the constraint, is one-to-one and such that $f(0) = q_0$.

From the constraints we get the following relation

$$h(q) = 0 \quad \Longrightarrow \quad \frac{\partial h}{\partial q}\frac{\partial q}{\partial \alpha} = 0 \quad \Longrightarrow \quad A^T \frac{\partial f}{\partial \alpha} = 0.$$

We shall use

$$J = \frac{\partial f}{\partial \alpha}.$$

We now reformat the dynamic equations of the robot in terms of the new coordinates α. Note that

$$\begin{aligned} f(\alpha) &= q, \\ J\dot{\alpha} &= \dot{q}, \qquad \text{and} \\ \dot{J}\dot{\alpha} + J\ddot{\alpha} &= \ddot{q}. \end{aligned}$$

Substituting the above in the dynamic equation 4.6 we get

$$M(q)J\ddot{\alpha} + (C(q,\dot{q})J + M(q)\dot{J})\dot{\alpha} + K_s f(\alpha) + K_f J\dot{\alpha} + A\lambda = \begin{bmatrix} \tau \\ 0 \end{bmatrix}.$$

Premultiplying the above equation by J^T we get the dynamic equation for the robot in terms of the new coordinates α:

$$\tilde{M}\ddot{\alpha} + \tilde{C}\dot{\alpha} + \tilde{K}_s f(\alpha) + \tilde{K}_f\dot{\alpha} = \tilde{\tau}, \tag{4.5}$$

where we have used

$$\begin{aligned}\tilde{M} &= J^T M J,\\ \tilde{C} &= J^T(CJ + M\dot{J}),\\ \tilde{K}_p &= J^T K_p,\\ \tilde{K}_f &= J^T K_f J,\\ \tilde{\tau} &= J^T \begin{bmatrix}\tau\\ 0\end{bmatrix}.\end{aligned}$$

□

We now prove the stability of the proposed joint PD based control law.

Theorem 4.1 (Stability of the joint PD controller) *Given $K_p, K_d > 0$ and that the stiffness of the manipulator is sufficient to apply the desired external force λ_0, the controller given in equation (4.4) asymptotically stabilizes the system given by*

$$\begin{aligned} M(q)\ddot{q} + C(q,\dot{q})\dot{q} + K_s q + K_f\dot{q} + A\lambda &= \begin{bmatrix}\tau\\ 0\end{bmatrix},\\ h(q) &= 0, \end{aligned} \tag{4.6}$$

to the desired position and applying the desired external force.

Proof: From the dynamic equation (4.3) we have

$$\begin{bmatrix}\tau_0\\ 0\end{bmatrix} = A_0\lambda_0 + K_s q_0.$$

The right-hand side in the above equation will yield the zeroes in the left-hand side at equilibrium. With the joint PD control law the equation of motion of the system becomes

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + K_s q + K_f\dot{q} + A\lambda = A_0\lambda_0 + K_s q_0 + K_p(q_0 - q) - K_d\dot{q}. \tag{4.7}$$

To prove stability of the control law we use the direct method of Lyapunov with the candidate function

$$\begin{aligned} V = \frac{1}{2}\dot{q}^T M\dot{q} + \frac{1}{2}(q-q_0)^T K_s(q-q_0) + \frac{1}{2}(q-q_0)^T K_p(q-q_0)\\ + \int_{q_0}^{q}(A\lambda_0 - A_0\lambda_0)^T dq. \end{aligned} \tag{4.8}$$

Consider the last term in the above equation.

$$\begin{aligned}\int_{q_0}^{q}(A\lambda_0 - A_0\lambda_0)^T dq &= \lambda_0(h(q) - h(q_0) - A_0(q-q_0)) \\ &= \lambda_0^T\Bigg(h(q_0) + A_0(q-q_0) + (q-q_0)^T \left.\frac{\partial^2 h}{\partial q^2}\right|_{q_0}(q-q_0) \\ &\quad + \mathcal{O}\left((q-q_0)^3\right) - h(q_0) - A_0(q-q_0)\Bigg) \\ &= \lambda_0^T\Bigg((q-q_0)^T \left.\frac{\partial^2 h}{\partial q^2}\right|_{q_0}(q-q_0) + \mathcal{O}\left((q-q_0)^3\right)\Bigg).\end{aligned}$$

The first three terms in the equation (4.8) are all positive definite. Hence the total function is positive definite irrespective of the sign of $\left.\frac{\partial^2 h}{\partial q^2}\right|_{q_0}$ if λ_0 is sufficiently small compared to $K_s + K_p$. Physically this means that to apply larger forces we require sufficiently large stiffness of the springs at the spring joints and a sufficiently large proportional gain.

The time derivative of the Lyapunov function is

$$\dot{V} = \dot{q}^T\left(M\ddot{q} + \frac{\dot{M}\dot{q}}{2} + K_s(q-q_0) + K_p(q-q_0) + A\lambda_0 - A_0\lambda_0\right).$$

Substituting for $M\ddot{q}$ from equation (4.7) in the above equation, the time derivative of the candidate Lyapunov function along the trajectories of the system becomes

$$\begin{aligned}\dot{V} = \dot{q}^T\Bigg(&A_0\lambda_0 + K_s q_0 + K_p(q_0 - q) - K_d\dot{q} - C\dot{q} - K_s q - K_f\dot{q} - A\lambda \\ &+ \frac{\dot{M}\dot{q}}{2} + K_s(q-q_0) + K_p(q-q_0) + A\lambda_0 - A_0\lambda_0\Bigg),\end{aligned}$$

which can be simplified as follows:

$$\begin{aligned}\dot{V} &= -\dot{q}^T(A\lambda) + \dot{q}^T(A\lambda_0) + \dot{q}^T\left(\frac{\dot{M}}{2} - C\right)\dot{q} - \dot{q}^T(K_f + K_d)\dot{q} \\ &= -\dot{q}^T(K_f + K_d)\dot{q},\end{aligned}$$

where we have used the dynamic equation of the robot, the nature of the constraints and the structural properties of the robot dynamic equation to simplify the obtained expression. Clearly

$$-\dot{q}^T(K_f + K_d)\dot{q} \leq 0$$

if $K_d + K_f$ is positive definite. Therefore, using the Lyapunov stability theorem we can say that the robot reaches an equilibrium state under the action of the applied control at which all the joint velocities are identically zero. We have to further prove that the equilibrium state obtained is the desired one. We shall use LaSalle's

invariance principle to prove convergence to the desired state.

It is convenient at this point to convert from the q coordinates to the generalized coordinates α in Lemma 4.1. At equilibrium at q_0, we have $\alpha = 0$, and from equation (4.5) the equilibrium torque is given by

$$\tilde{K}_s f(0) = J^T K_s f(0).$$

In the new coordinates therefore the control law is

$$\tilde{\tau} = J^T K_s f(0) + J^T K_p\big(f(0) - f(\alpha)\big) - J^T K_d J\dot{\alpha}. \tag{4.9}$$

At equilibrium under the control law $\dot{q}, \ddot{q} = 0$ and therefore $\dot{\alpha}, \ddot{\alpha} = 0$. Now we apply LaSalle's principle to the dynamic equations in the new coordinates and try to find the largest invariant set

$$\begin{aligned} J^T K_s f(\alpha) &= J^T K_s f(0) + J^T K_p\big(f(0) - f(\alpha)\big) \\ \Longrightarrow\ & J^T(K_s + K_p)(f(\alpha) - f(0)) = 0 \end{aligned}$$

Using a Taylor series expansion about $\alpha = 0$ we get

$$\begin{aligned} J^T(K_s + K_p)(f(\alpha) - f(0)) &= J^T(K_s + K_p)(f(0) + J\big|_{\alpha=0}\alpha + \mathcal{O}\left(\alpha^2\right) - f(0)) \\ &= J^T(\alpha)(K_s + K_p)J\big|_{\alpha=0}\alpha + \mathcal{O}\left(\alpha^2\right) \\ &= J^T\big|_{\alpha=0}(K_s + K_p)J\big|_{\alpha=0}\alpha + \mathcal{O}\left(\alpha^2\right) \\ &= 0. \end{aligned}$$

In the above we note that $K_s + K_p$ is full rank and $J(0)$ has full column rank. Therefore using the implicit function theorem and ignoring the higher order terms in α we can say that $\alpha = 0$ is the unique solution to the equilibrium conditions. Thus the largest invariant set consists of the desired equilibrium point and the controller will therefore asymptotically stabilize the finger to the desired equilibrium point. □

There are some points to be noted about the above proof. It is a local proof and it does not depend on only the last link being flexible. Therefore it will also hold for a robot with both the links flexible. Further, the same proof holds independent of the number of sub-links into which the flexible link is divided.

The simulation data presented in this chapter is for the setup shown in Figure 4.3. The length of the base link in the figure is 10 cm and the sublinks modeling the flexible link are each a third of that. The initial and final position of the manipulator are shown in Figure 4.4. Note that the manipulator undergoes significant flexible distortion at the final position due to application of the external force.

Simulation results for this controller are presented in Figure 4.5. The control law converged either to the desired equilibrium point or to some other equilibrium point depending on the initial conditions applied to the finger (as noted, the proof only guarantees local stability). An example of each is given in Figure 4.5. In both the simulations the properties and the parameters of the controller and the finger

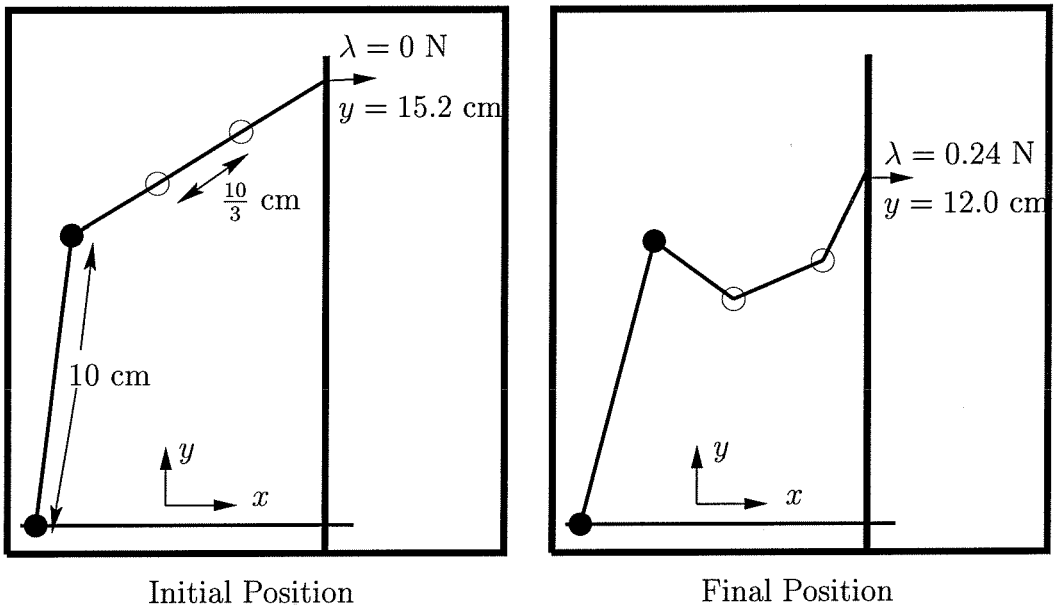
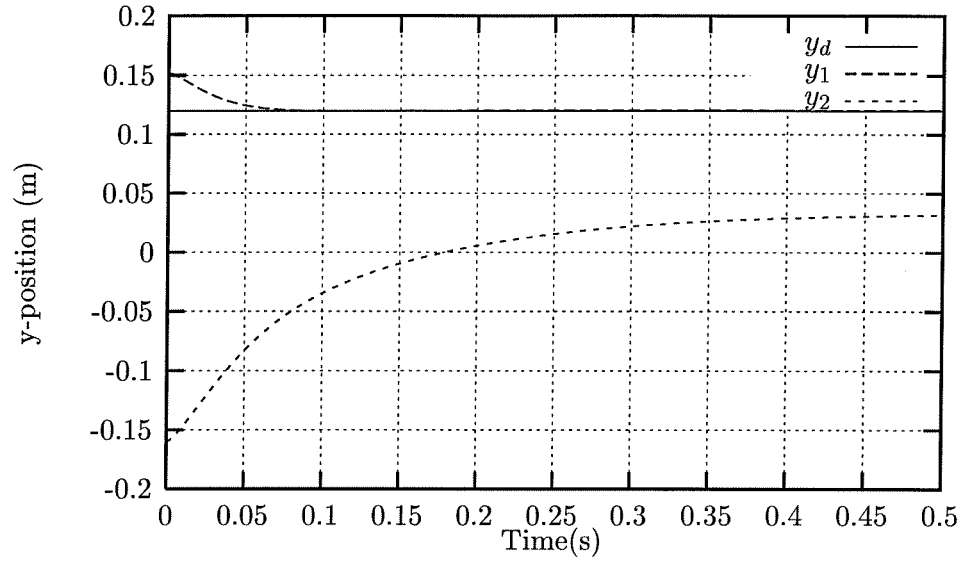


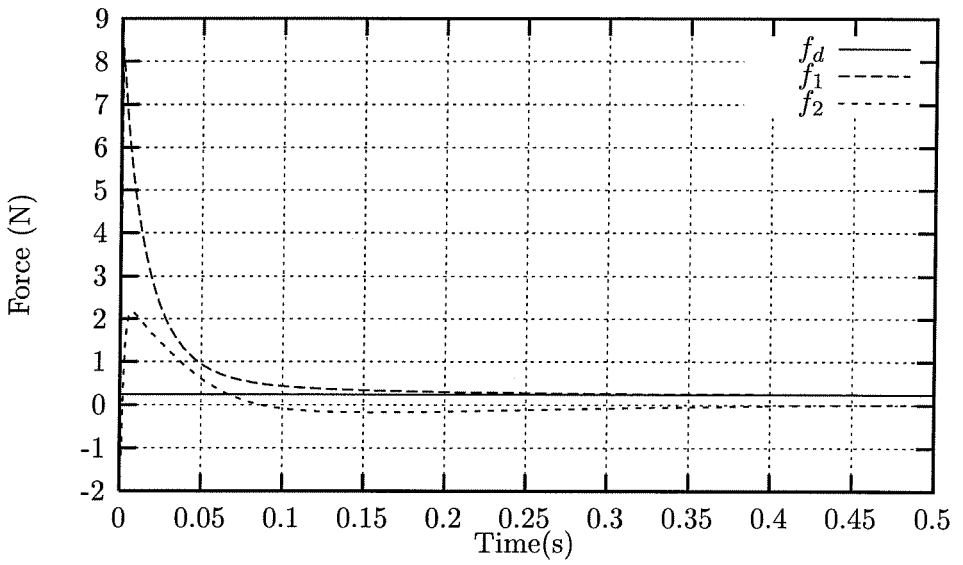
Figure 4.4 Simulation task for controller task.

were exactly the same. The only difference was the initial point from which each simulation was started. In the simulation data we see the contact force achieving negative values, which are clearly not possible in the real case. This is due to the particular way in which the simulation enforces the constraint condition.

The experiments for a finger pushing against a wall were done using a planar, two degree of freedom finger with the last link flexible and pushing against a wall instrumented with a force/torque sensor. For the experimental run of a finger pushing against a wall with the joint PD controller, the equilibrium position values were determined by using a simple joint PD based controller to push against the wall and noting the values of the torques and the joint angles at equilibrium. These values were then used by the joint PD and feedforward force controller. Figure 4.6 shows two runs (one from either side of the desired equilibrium position on the wall) of the controller. The control on position is better than the control on the force applied. The error in the desired force during experimentation could be due to the zero friction assumption in the analysis and the simulations. In the experimental setup we could not eliminate friction totally and the action of pushing against the wall resulted in significant friction. The other factors which contribute to the error in tracking the force are inherent in the tendon actuation scheme. It is very difficult to model the friction present due to the sliding of the tendons on various surfaces. In addition the coupling matrix holds for a limited range of configurations and gives rise to inaccuracies in the torque being transmitted to the joints during experimentation. As previously noted the practicality of this controller is limited because equilibrium configurations and torques must be precomputed.

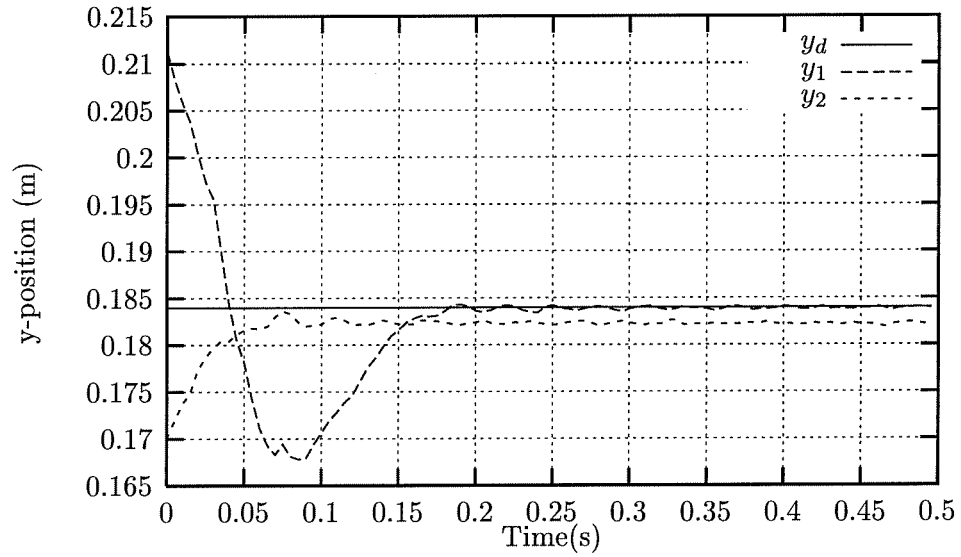


(a) Position Regulation

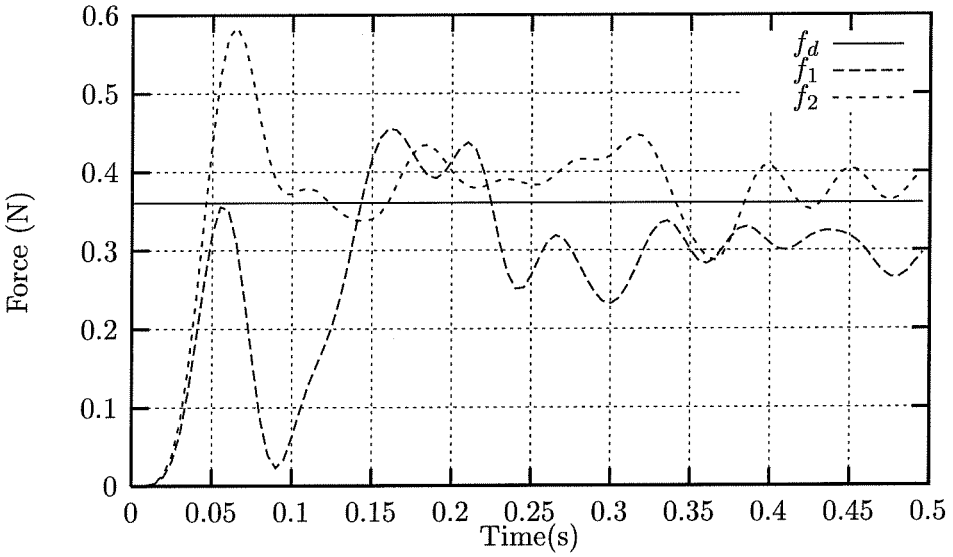


(b) Force Regulation

Figure 4.5 Simulation results for joint PD with feedforward force.



(a) Position Regulation



(b) Force Regulation

Figure 4.6 Experimental results for joint PD with feedforward force.

4.3 Controller Ideas from Analysis of the Reduced System

The reduced system was developed from the singular perturbation analysis of the preceding chapter. We rewrite the equations (3.26), (3.27) and (3.28) here for ease of referral. The reduced order system which we will use as the centerpiece of the following analysis is

$$\begin{aligned} m_{11}\ddot{\Theta} + C_{11}\dot{\Theta} + k_{f\theta}\dot{\Theta} + A_\theta\lambda &= \tau \\ k_{s\psi}\Psi + A_\psi\lambda &= 0 \\ h(\Theta,\Psi) &= 0. \end{aligned} \tag{4.10}$$

We discuss two controllers in this section. Recall that we wish to regulate the end-position of the manipulator to the desired point (X_0) and to apply a desired force(λ_0) against the constraint at the desired point. Both the goals are prescribed in the workspace and must be consistent with the constraints on the system. In what follows we use K_p to represent a positive definite, constant, diagonal matrix of proportional gains and K_d to represent a positive definite, constant, diagonal matrix of derivative gains. We also use the subscript 0 to denote quantities at the final equilibrium state. The stability of the joint PD controller described in the previous section can also be proved in the singular perturbation framework. We will however not present that proof here.

We introduce the following additional notation. In the rest of this chapter we shall represent the Jacobian mapping as

$$J_{\alpha\beta} = \frac{\partial\alpha}{\partial\beta}.$$

Further, we will denote the mapping between the configuration space and the workspace by

$$g(\Theta,\Psi) : \ q \mapsto X.$$

As usual, the constrained finger can be treated as a free finger with additional forces applied to ensure that the constraint conditions are met. Denoting the constraint in the workspace by

$$s(X) = 0,$$

we have

$$s(g(q)) = 0,$$

which is the same as $h(q) = h(\Theta,\Psi) = 0$. The generalized forces of constraint are always in the direction orthogonal to the constraint, which is given by ∇h. In our

case the torques generated by the constraint surface (the wall) are given by

$$\tau_n = (\nabla h)\lambda = (J_{sX} J_{Xq})^T \lambda = A\lambda,$$

where λ is the Lagrange multiplier to be found from the dynamic equation. In this notation we can also write

$$\begin{aligned} A_\theta &= \frac{\partial h}{\partial \Theta}^T = (J_{sX} J_{X\Theta})^T \quad \text{and} \\ A_\psi &= \frac{\partial h}{\partial \Psi}^T = (J_{sX} J_{X\Psi})^T. \end{aligned} \tag{4.11}$$

In the actual system we expect to have sensors for the location of the end-point of the manipulator and for the applied force at the tip. The experimental setup on which these controllers were tried out were instrumented to provide this data in addition to the data on joint angles (refer to Appendix A).

4.3.1 The J_* controller

The reduced system is completely known by knowing the Θ variables as was shown in equation (3.29). The Jacobian between the configuration space and the workspace can be derived as follows for the reduced order system

$$X = g(\Theta, \Psi(\Theta))$$

therefore,

$$\dot{X} = J_{X\Theta}\dot{\Theta} + J_{X\Psi}\dot{\Psi} = (J_{X\Theta} + J_{X\Psi} J_{\Psi\Theta})\dot{\Theta} = J_*\dot{\Theta}. \tag{4.12}$$

Note that J_* is a square matrix for a manipulator with number of actuated joints equal to the the number of workspace coordinates (which is the case we are dealing with) and we assume it is invertible in what follows (this is a reasonable assumption). In a real system we can compute J_* online from the knowledge of Θ and λ using the flexible-sublink model. Note further that J_{sX} is known and $J_{X\Theta}$ can be computed from the knowledge of the manipulator tip and the positions of its actuated joints.

Consider the workspace control law

$$\tau = J_*^T(K_p(X_0 - X) - K_d\dot{X}) + \tau_0, \tag{4.13}$$

where

$$\tau_0 = \left(J_{sX_0} J_{X\Theta_0}\right)^T \lambda_0. \tag{4.14}$$

This is a workspace PD control law with the transpose of the Jacobian J_* being used to map the workspace forces to the joint space. In addition the law does a feedforward of the constraint torque required at the final equilibrium. We will discuss this in more detail later.

Theorem 4.2 (Reduced system stability with J_* control law) *Assuming K_p, $K_d > 0$, the controller given in equation (4.13) asymptotically stabilizes the reduced system given by equation (4.10) to the desired end-point position and applying the desired external force.*

Proof: **(Reduced system.)** We rewrite the system with its controller as

$$\begin{aligned} m_{11}\ddot{\Theta} + C_{11}\dot{\Theta} + k_{f\theta}\dot{\Theta} + (J_{sX}J_{X\Theta})^T\lambda &= \tau \\ &= J_*^T(K_p(X_0 - X) - K_d\dot{X}) + (J_{sX_0}J_{X\Theta_0})^T\lambda_0, \\ h(\Theta) &= 0. \end{aligned} \tag{4.15}$$

To prove stability we use a Lyapunov approach with the candidate Lyapunov function

$$\begin{aligned} V = \frac{\dot{\Theta}^T m_{11}\dot{\Theta}}{2} + \frac{(X - X_0)^T K_p (X - X_0)}{2} + \frac{(\Psi - \Psi_0)^T k_{s\psi}(\Psi - \Psi_0)}{2} \\ + \int_{\Theta_0}^{\Theta} \Big((J_{sX}J_*)^T\lambda_0 - (J_{sX_0}J_{X\Theta_0} + J_{sX_0}J_{X\Psi_0}J_{\Psi\Theta})^T\lambda_0 \Big)^T d\Theta. \end{aligned} \tag{4.16}$$

The positive definiteness of this Lyapunov function is evident except for the last integral term. However we show that this term is identical to the integral term in Lyapunov function in the proof of stability of the joint PD controller. The integral term in equation (4.8) is

$$\int_{q_0}^{q} (A\lambda_0 - A_0\lambda_0)^T dq,$$

in which we can use

$$dq = \begin{bmatrix} d\Theta \\ d\Psi \end{bmatrix} = \begin{bmatrix} d\Theta \\ \frac{\partial\Psi}{\partial\Theta} d\Theta \end{bmatrix} = \begin{bmatrix} I \\ \frac{\partial\Psi}{\partial\Theta} \end{bmatrix} d\Theta = \begin{bmatrix} I \\ J_{\Psi\Theta} \end{bmatrix} d\Theta,$$

and we can write from equation (4.11)

$$A = \begin{bmatrix} A_\theta \\ A_\psi \end{bmatrix} = \begin{bmatrix} (J_{sX}J_{X\Theta})^T \\ (J_{sX}J_{X\Psi})^T \end{bmatrix}.$$

Substituting the above into the integral we get the integral term of equation (4.16). The time derivative of the function along the system trajectory is

$$\begin{aligned} \dot{V} = \dot{\Theta}^T \Big(m_{11}\ddot{\Theta} + \frac{\dot{m}_{11}\dot{\Theta}}{2} + (J_{sX}J_*)^T\lambda_0 - (J_{sX_0}J_{X\Theta_0} + J_{sX_0}J_{X\Psi_0}J_{\Psi\Theta})^T\lambda_0 \Big) + \\ \dot{X}^T K_p (X - X_0) + \dot{\Psi}^T k_{s\psi}(\Psi - \Psi_0). \end{aligned} \tag{4.17}$$

We simplify the above by noting some relations between quantities involved. First we have from the constraint,

$$s(X) = 0 \quad \Longrightarrow \quad \dot{X}^T J_{sX}^T = 0 \quad \Longrightarrow \quad \dot{\Theta}^T (J_{sX} J_*)^T = 0. \tag{4.18}$$

Also,

$$\begin{aligned}
&\dot{\Theta}^T \left(m_{11}\ddot{\Theta} + \frac{\dot{m}_{11}\dot{\Theta}}{2} - (J_{sX_0} J_{X\Theta_0})^T \lambda_0 \right) + \dot{X}^T K_p (X - X_0) \\
&\quad = \dot{\Theta}^T \Bigg(J_*^T K_p (X_0 - X) - J_*^T K_d \dot{X} + (J_{sX_0} J_{X\Theta_0})^T \lambda_0 - C_{11}\dot{\Theta} + \\
&\qquad \frac{\dot{m}_{11}\dot{\Theta}}{2} - k_{f_\Theta}\dot{\Theta} - (J_{sX} J_{X\Theta})^T \lambda + J_*^T K_p (X - X_0) - \\
&\qquad (J_{sX_0} J_{X\Theta_0})^T \lambda_0 \Bigg) \\
&\quad = -\dot{X}^T K_d \dot{X} - \dot{\Theta}^T K_{f_\Theta} \dot{\Theta} - \dot{\Theta}^T (J_{sX} J_{X\Theta})^T \lambda,
\end{aligned} \tag{4.19}$$

and

$$\dot{\Psi}^T k_{s\psi} (\Psi - \Psi_0) = \dot{\Theta}^T J_{\Psi\Theta}^T k_{s\psi} (\Psi - \Psi_0) = \dot{X}^T J_*^{-T} J_{\Psi\Theta}^T k_{s\psi} (\Psi - \Psi_0). \tag{4.20}$$

Substituting into equation (4.17) we get,

$$\begin{aligned}
\dot{V} = -\dot{X}^T K_d \dot{X} - \dot{\Theta}^T K_{f_\Theta} \dot{\Theta} - \dot{\Theta}^T \big(&(J_{sX} J_{X\Theta})^T \lambda - (J_{sX_0} J_{X\Psi_0} J_{\Psi\Theta})^T \lambda_0 + \\
&J_{\Psi\Theta}^T k_{s\psi} (\Psi - \Psi_0) \big).
\end{aligned}$$

The following relations are true in the reduced system (and follow from the reduced system equations):

$$\begin{aligned}
(J_{sX} J_{X\Psi})^T \lambda &= -k_{s\psi} \Psi, \\
(J_{sX_0} J_{X\Psi_0})^T \lambda_0 &= -k_{s\psi} \Psi_0 \\
(J_{sX} J_{X\Theta})^T \lambda &= J_{\Psi\Theta}^T k_{s\psi} \Psi.
\end{aligned} \tag{4.21}$$

Therefore,

$$\begin{aligned}
-(J_{sX} J_{X\Theta})^T \lambda - (J_{sX_0} J_{X\Psi_0} J_{\Psi\Theta})^T \lambda_0 + J_{\Psi\Theta}^T k_{s\psi} (\Psi - \Psi_0) \\
&= -J_{\Psi\Theta}^T k_{s\psi} \Psi + J_{\Psi\Theta}^T k_{s\psi} \Psi_0 + J_{\Psi\Theta}^T k_{s\psi} \Psi - J_{\Psi\Theta}^T k_{s\psi} \Psi_0 \\
&= 0,
\end{aligned}$$

giving,

$$\dot{V} = -\dot{X}^T K_d \dot{X} - \dot{\Theta}^T K_{f_\Theta} \dot{\Theta}.$$

The derivative of the Lyapunov function is therefore negative semi-definite. Using Lyapunov's theorem we can say that the manipulator reaches an equilibrium under the action of the control law. To prove asymptotic stability we use LaSalle's theo-

rem, similarly to the case of the joint PD controller. □

The above theorem proves that the J_* controller is successful in stabilizing the reduced system to the desired position and applying the desired external force. The behavior of the full system under this control law must now be determined. This follows readily from what has been presented so far and is formalized in the following theorem.

Theorem 4.3 (Full system behavior with J_* control) *For small ϵ and assuming $K_p, K_d > 0$, the J_* control law described in equation* (4.13)

$$\tau = J_*^T(K_p(X_0 - X) - K_d\dot{X}) + \tau_0,$$

with

$$\tau_0 = \left(J_{sX_0} J_{X\Theta_0}\right)^T \lambda_0,$$

causes the response of the constrained flexible manipulator system given by equation (3.24)

$$\begin{bmatrix} m_{11} & m_{12} \\ \epsilon^2 m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} \ddot{\Theta} \\ \epsilon^2\ddot{\Psi} \end{bmatrix} = \begin{bmatrix} \tau \\ 0 \end{bmatrix} - \left(\begin{bmatrix} c_{11}\dot{\Theta} & \epsilon^2 c_{12}\dot{\Theta} \\ \epsilon^2 c_{21}\dot{\Psi} & \epsilon^2 c_{22}\dot{\Psi} \end{bmatrix} \begin{bmatrix} \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & k_{s\psi} \end{bmatrix} \begin{bmatrix} \Theta \\ \Psi \end{bmatrix} + \begin{bmatrix} k_{f\theta} & 0 \\ 0 & \epsilon k_{f\psi} \end{bmatrix} \begin{bmatrix} \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} + \begin{bmatrix} A_\theta \\ A_\psi \end{bmatrix} \lambda \right), \quad (4.22)$$

to remain within $\mathcal{O}(\epsilon)$ of the reduced system which is asymptotically stabilized to the desired position and applying the desired force.

Proof: We proved in Theorem 3.4 that the boundary layer subsystem of the above system, equation (4.22), obtained by setting ϵ to zero, was asymptotically stable. This being an autonomous system, this is equivalent to proving uniform asymptotic stability for the system. The conditions set out in Theorem 3.1 are therefore satisfied, hence assuring the existence of the reduced system for which Theorem 4.2 proves uniform asymptotic stability with the J_* controller. Restricting Θ to a compact set and using Theorem 3.2 we can therefore state that the response of the full system will remain within $\mathcal{O}(\epsilon)$ of the response of the reduced system and for $\epsilon = 0$ (the reduced system) the J_* control law will asymptotically stabilize the system. □

Remarks on system behavior: To prove asymptotic stability for the full system we need to use Theorem 3.3. We can prove exponential stability of the boundary layer system (by using results in Murray, Li and Sastry [40]). For the autonomous system uniformity in time follows directly. Uniformity in Θ is achieved by restricting Θ to a compact set. Uniform asymptotic stability of the reduced system under the J_* control law has been proved. Using converse Lyapunov theorems (refer to Khalil [23]) it is immediately true that there exist Lyapunov functions $V(\Theta)$ and

$W(\Theta, \Phi)$ (refer to Theorems 3.3 and 3.4) which satisfy the first three and fifth conditions of Theorem 3.3. The fourth condition is harder to prove. It requires knowledge of the Lyapunov function $V(\Theta)$. Note that we do not have the Lyapunov function V as we could only prove stability by invoking LaSalle's principle. □

In an actual experimental or simulation setup to avoid precomputation of the final constraint torque we use $(J_{sX}J_{X\Theta})^T\lambda_0$ instead of τ_0 in equation (4.14). It might seem at first glance that $(J_{sX}J_*)^T\lambda_0$ can be used instead of τ_0. However, since we derive J_* from the constraint equations its effect is restricted to the unconstrained directions. Therefore, its transpose cannot be used to transform static forces against the constraint, between the workspace and the joint space (like Jacobians for fully actuated non-redundant manipulators can). These issues are basic to the Jacobian mapping and a discussion of these is beyond the scope of this thesis.

From the last two relations in equation (4.10) (which define the reduced manifold) we can write

$$\begin{bmatrix}\dot{\lambda}\\ \dot{\Psi}\end{bmatrix} = -\begin{bmatrix}(J_{sX}J_{X\Psi})^T & k_{s\psi}+\frac{\partial(J_{sX}J_{X\Psi})^T}{\partial\Psi}\lambda\\ 0 & J_{sX}J_{X\Psi}\end{bmatrix}^{-1}\begin{bmatrix}\frac{\partial(J_{sX}J_{X\Psi})^T}{\partial\Theta}\lambda\\ J_{sX}J_{X\Theta}\end{bmatrix}\dot{\Theta}$$
$$= \begin{bmatrix}J_{\lambda\Theta}\\ J_{\Psi\Theta}\end{bmatrix}\dot{\Theta}. \tag{4.23}$$

As J_{sX} is known and $J_{X\Theta}$ can be computed we can compute J_* from the above. Figure 4.7 shows the simulation results for the J_* controller. We could not perform experiments with the J_* controller because of drift in our force sensor. However, we present detailed simulation results in the next chapter.

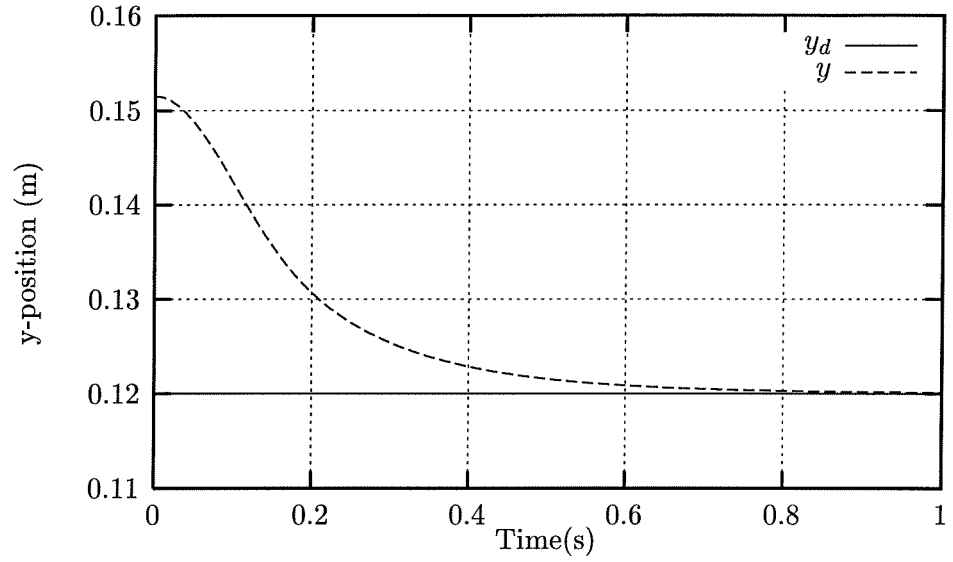
4.3.2 The instantaneous Jacobian controller

We call $J_{X\Theta}$ the instantaneous Jacobian of the flexible manipulator because it is the Jacobian of a rigid link robot with the same number of links as the original, with joints at the current actuated joint positions of the flexible robot, and endpoint coinciding with the endpoint of the flexible robot. $J_{X\Theta}$ is a Jacobian computable once the endpoint and the joint positions of the manipulator are known. Not only is the shape of the flexible beam not required, neither is a model of the flexibility required.

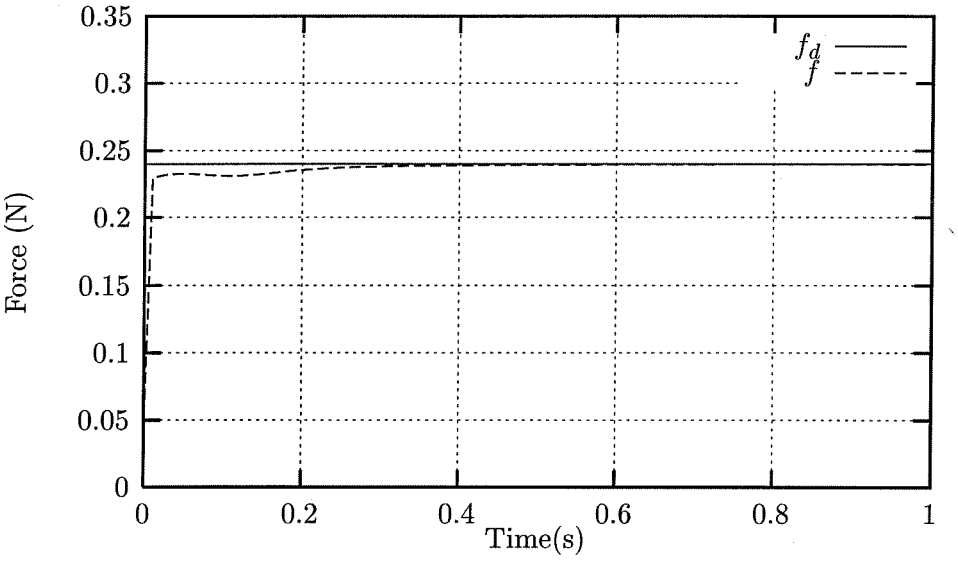
Consider a control method in which the instantaneous Jacobian is constructed at each configuration of the finger based on feedback of the endpoint and joint positions of the finger. It is then used to calculate the joint torques required to apply the desired force against the wall and to move the tip towards the goal point. The control law is given by

$$\tau = J_{X\Theta}^T(K_p(X_0 - X) - K_d\dot{X} + J_{sX}\lambda_0). \tag{4.24}$$

Figure 4.8 shows the motivation for the control law. At each instant the actual robot is replaced by a "virtual rigid robot" (shown by the dotted line in the figure)



(a) Position Regulation



(b) Force Regulation

Figure 4.7 Simulation results for the $\mathbf{J}_*$ controller.

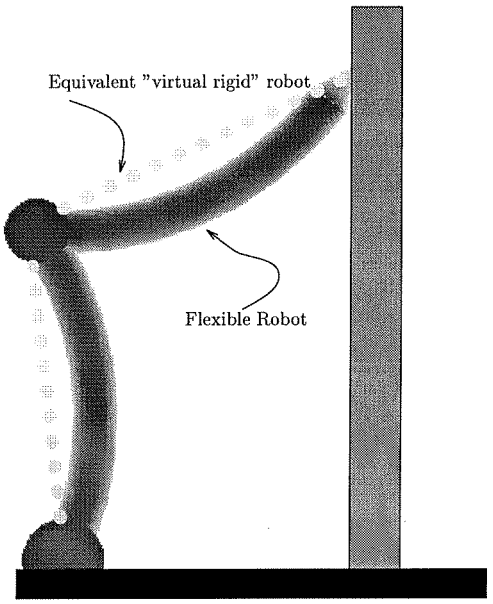


Figure 4.8 Motivation for a instantaneous Jacobian based controller.

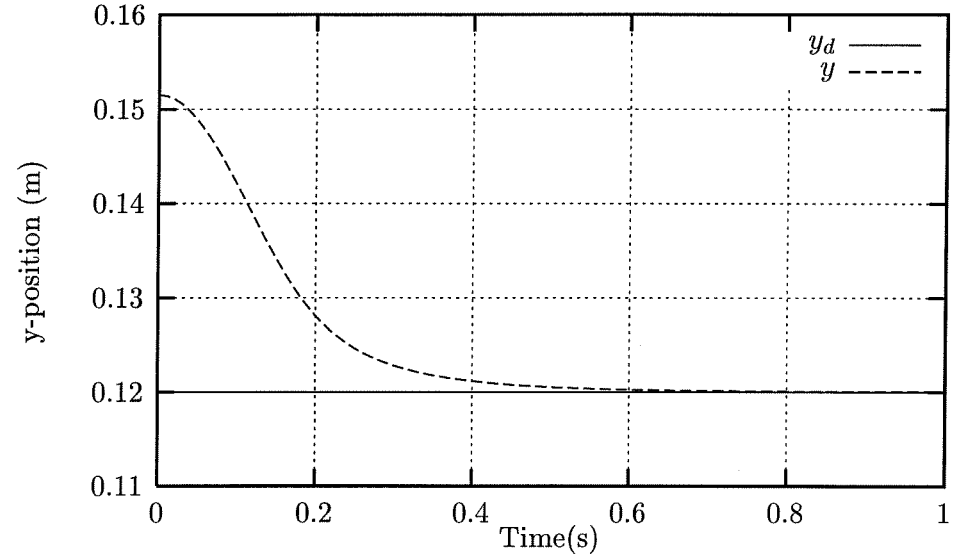
and the torques are calculated based on that. Note that in any robot the application of joint torques based on the Jacobian calculation does not require the shape of the links to be known. Knowledge of the positions of the joints suffices. Therefore, there is some intuitive sense in this control law. Recall also the discussion on the reduced system presented at the end of the previous chapter. If the flexible shape change happens so fast as to be almost instantaneous and stabilizes itself, there is little to be gained from trying to control the transient, especially in the face of controller bandwidth limitations.

In simulations and in actual experimental tests a controller based on this Jacobian was found to work very well. However, we could prove its stability only after making additional assumptions. We have

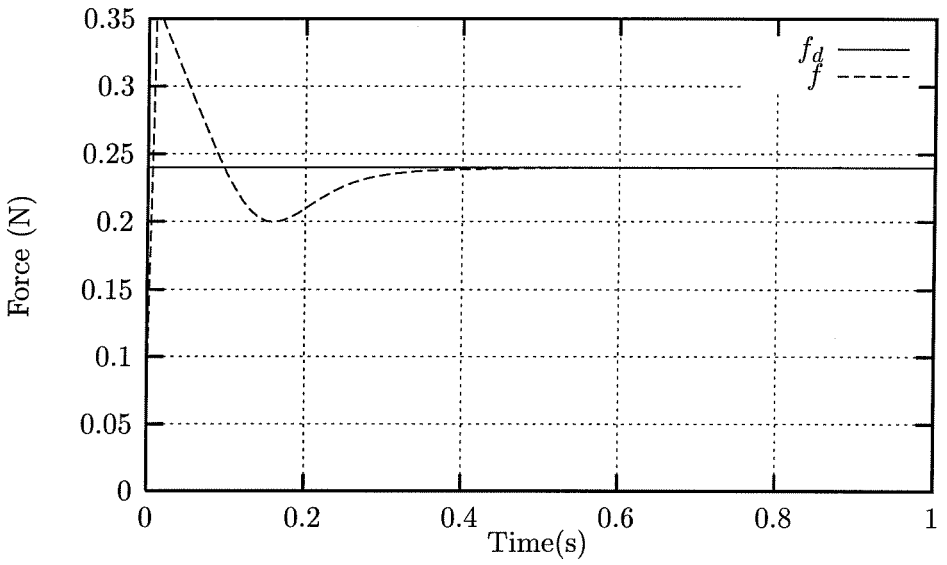
$$J_* = J_{X\Theta} + J_{X\Psi} J_{\Psi\Theta}$$

from equation (4.12). If we assume that the contribution from $J_{X\Psi} J_{\Psi\Theta}$ is small compared to that of $J_{X\Theta}$ in the reduced system, under the action of the instantaneous Jacobian based control law then we can ignore this term. In that case the proof of stability of the instantaneous Jacobian based control law is identical to the one for J_*.

Figure 4.9 shows the simulation results for the instantaneous Jacobian based controller. The instantaneous Jacobian based controller is particular easy to implement in an experimental setup. We present experimental data for a instantaneous Jacobian controlled flexible finger pushing against a wall in Figure 4.10.

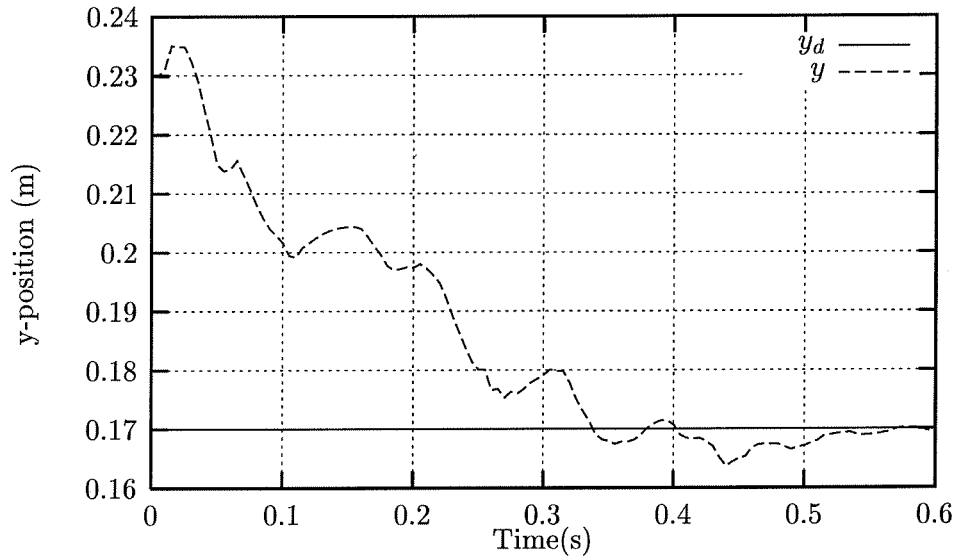


(a) Position Regulation

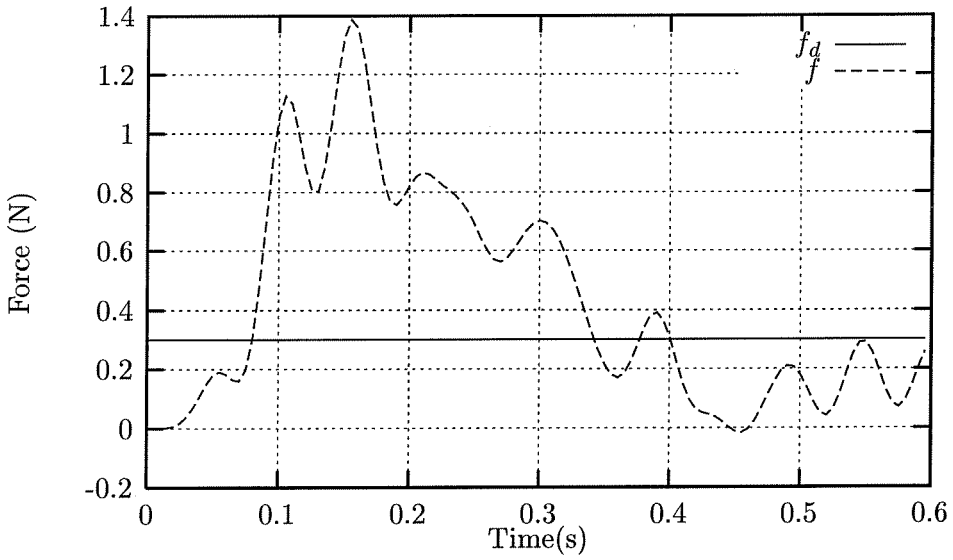


(b) Force Regulation

Figure 4.9 Simulation results for the instantaneous Jacobian controller.



(a) Position Regulation



(b) Force Regulation

Figure 4.10 Experimental results for the instantaneous Jacobian controller.

4.4 Implementation Issues

Mathematical analysis and simulations, no matter how detailed, cannot capture all the subtleties of the real problem. Further, there is the issue of an implementable control law for a real system. A control idea may exhibit very desirable properties on paper, but if it cannot be implemented on a real system its usefulness for engineering purposes is severely limited. In this section we consider issues of implementation of the controllers developed earlier in this chapter.

The joint PD controller clearly has limited applicability due to the requirement of precalculation. If we cannot perform the calculation in real-time, the only way of implementation is to set up lookup tables calculated offline. In certain applications this may not be a very bad solution. For example, in the automotive industry robots which paint cars often have a training mode in which they are "led" by hand to store the painting trajectory. During regular operation the robot retraces the trajectory from data stored during the training run. A similar approach could very well work for the joint PD based controller. The method is attractive in itself, because barring the precalculation it is a control law which can be readily implemented with existing hardware. From our simulations it also appears that this controller is the fastest to converge to the desired state. We delay the discussion of those issues till the next chapter.

The J_* controller does not have the problem of precalculation which the joint PD controller does. It does require more sensing. The types of sensing devices required for feedback to this control law do exist, therefore this law is implementable in real setups. This is the only one of the controllers which depends on the model of the flexible beam. For the sublink model which we have used, equation (4.23) can be used to calculate J_*. We have also discussed previously the procedure for identifying parameters for our modeling approach. Also note that for the joint PD approach, if we want to have a real-time computation of the precalculations then we do need to have a model for the flexibility.

The instantaneous Jacobian controller does not have the precalculation problem and at the same time requires less sensing than the J_* controller. It does not depend on a model of beam flexibility for calculating the control. Thus, it is possible that the instantaneous Jacobian-based controller will be able to control fingers made of non-metallic material which may be hard to model. The sensing of the end-points and the joints can be done in various ways. One possible approach, using fairly new electronic technology, is described in the chapter on experimentation. However, there does exist literature on the estimation of the end-point position and orientation of flexible links using strain gages [42]. Inspite of these perceived advantages, the lack of a formal proof of stability is a major hindrance to its application.

We summarize these observations in Table 4.1. We point out that the instantaneous Jacobian and J_* can be used in place of the usual robot Jacobian in any workspace control law. Their use is not limited to the PD control law we have dealt with in this chapter. Their applicability is therefore very general and should be considered an augmentation or modification of existing workspace control laws for rigid robots for the control of flexible robots.

Controller	Control calculations	Sensors	Remarks
Joint PD with feedforward force	Equilibrium joint torques(actuated), equilibrium joint angles(actuated)	Angle encoders	Training mode to avoid precalculation
$\mathbf{J}_*$	J_* from flexibility model	Angle encoders, endpoint force sensors	Requires flexibility model
Instantaneous Jacobian	Standard Jacobian calculation	Actuated joint position sensors	Model independent

Table 4.1 Summary of implementation issues for flexible robot controllers.

Chapter 5

Parametric Studies using Numerically Simulated System

Simulations provide a bridge between mathematical analysis and experimentation. They can be used to identify system properties not evident from the analysis and difficult to implement in experiments. In addition, in the case of control applications, it is important to know how precisely the system to be controlled must adhere to the assumptions made during analysis for the control laws developed as a result of the analysis to hold. This can be determined much more easily in simulations where we have strict control on the parameters affecting the system. In this chapter we present results of simulations of the exact system analyzed. The aims of the simulation are to determine the properties of the flexible manipulator system as well as to test our assumptions. In addition we gain an insight into the tradeoffs involved in using flexible manipulators for constrained motion tasks.

5.1 Simulation Setup and Aim of Simulation

For all the simulations carried out, the model is the same as that used in the previous chapter. The simulation was therefore of the exact system which has been analyzed. The dynamic model of the manipulator is derived considering the mass of the links and the sub-links to be concentrated at their midpoints.

The constraints were were enforced using the so-called "Baumgarte method" [1]. We briefly describe the method here. To incorporate the effect of a holonomic constraint, $h(q,t) = 0$, we differentiate the constraint twice to get a second order differential equation $\ddot{h} = 0$ which can then be incorporated into the differential equation using Lagrange multipliers. This is what we did previously when deriving the equation of motion of a constrained manipulator. Numerical integration of these equations without further imposition of the constraints is unstable, in that if the integration yields $h = \delta, \dot{h} = \epsilon$, at the nth step (due to simulation noise and precision errors), then at the next time step (assuming a simulation step of t seconds), we can expect $h = \epsilon t + \delta$. This behavior will in general not be compensated and the constraint will be allowed to "drift." One possibility of getting around this is to use a differential-algebraic equation (DAE) solver and solve the algebraic constraint simultaneously. These are however difficult to implement and much slower

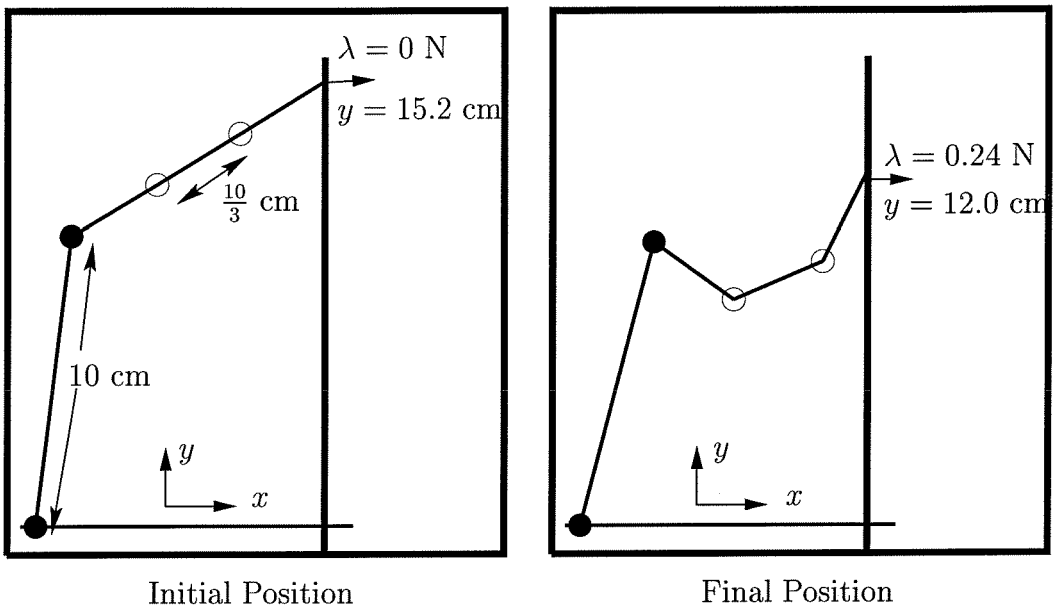


Figure 5.1 Simulation task for controllers.

in execution. In Baumgarte's method a PD control is imposed on the constraint:

$$\ddot{h} = -\alpha\dot{h} - \beta h.$$

For $\alpha, \beta > 0$ this serves to decrease the error in the constraint. Baumgarte in [1] proves that the original equations are not modified by the introduction of these "computational control terms."

The dynamic equations themselves are quite lengthy and complicated, as we are essentially simulating a four link robot, and there are a large number of cross-coupling terms. Hence, the dynamics were calculated using a symbolic mathematics package. The initial and final task for the controllers were identical to that for the simulations in the last chapter and are reproduced in Figure 5.1 for ease of reference. The simulations were programmed in C and performed on various SUN Sparc based UNIX workstations.

The analysis in the foregoing chapters was done under certain assumptions made about the nature of the dynamics of a flexible manipulator system performing constrained motion tasks. The successful implementation in our experiments of the control laws developed as a result of the analysis points towards the essential soundness of the approach and the practicability of the ensuing control laws. These results are presented in the next chapter. Our simulations are intended to supplement the experimental results. They are partly an effort to determine the behavior of flexible manipulators performing tasks requiring interaction with the environment, as very little data is available in the literature which describes such systems. The simulations also enable us to characterize the different controllers by their behavior and their performance under varying conditions.

5.2 Simulation Data

We have presented simulation data in the previous chapter for the controllers we proposed. In this section we take a closer look at the dynamic behavior and performance of the controllers. Unless otherwise stated, the final conditions for the joint PD controller are determined from the final state of the instantaneous Jacobian controller performing the same task.

5.2.1 Configurations of the flexible manipulator

Even when performing the same task the controllers execute significantly different control action. This is made evident when we consider the intermediate configurations of the manipulator between the same initial and final states. We present the configurations undergone by the flexible manipulator in Figures 5.2, 5.3 and 5.4. These correspond respectively to the simulation results presented in the previous chapter, Figures 4.5, 4.7 and 4.9. Figure 5.2 shows the sequence of configurations for the joint PD based controller. Compared to the other simulations it is evident that this controller achieves its final position very rapidly. Also notable is the fact that while the joint PD based controller starts flexing the flexible link almost from the beginning of its downward motion, the other two controllers seem to apply the compression after the end point has been achieved. This is partly due to the greater accelerations that the manipulator is subjected to by the joint PD controller. Another interesting thing to note is that the configuration of the manipulator changes even after the final position is achieved. These are internal motions of the manipulator, made possible due to the extra degrees of freedom bestowed by the flexibility. The J_* and the instantaneous Jacobian based controllers were simulated with identical gains. This might explain the similarity of the configurations of the manipulator while being controlled by either. However, note that they behave significantly differently in terms of the forces they apply against the constraint (Figures 4.7 and 4.9). These observations must be discussed within a larger framework, and we will discuss these further after we present other observations from simulations.

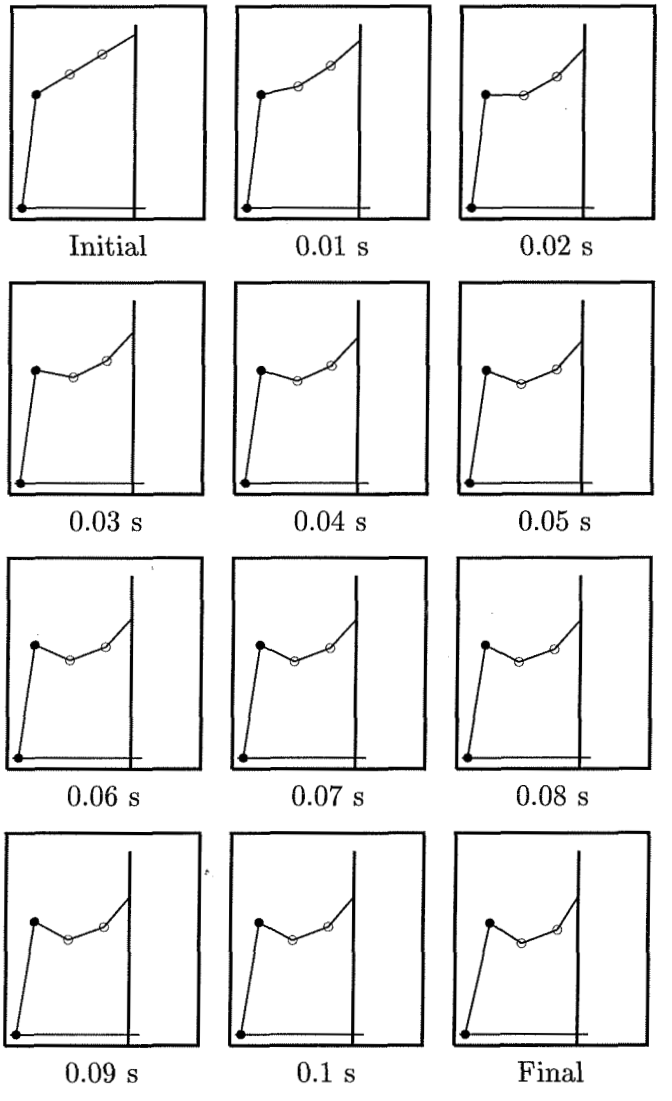


Figure 5.2 Joint PD controller ($m = 0.005$ kg, $k = 0.01$ N/rad).

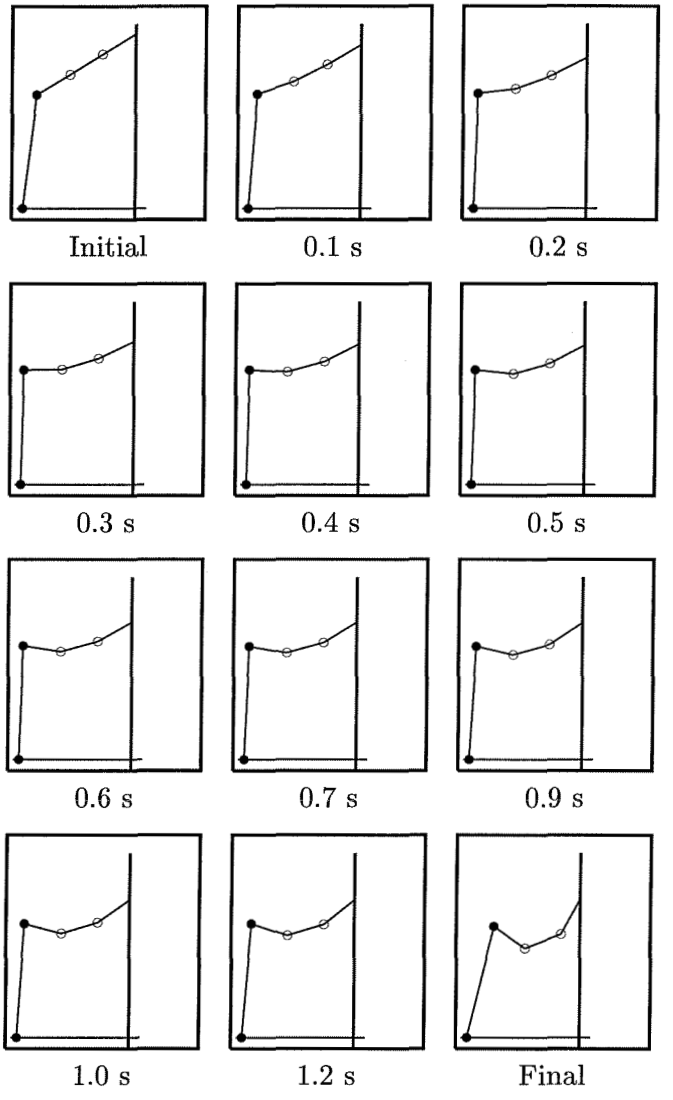


Figure 5.3 $\mathbf{J}_*$ controller ($m = 0.005$ kg, $k = 0.01$ N/rad).

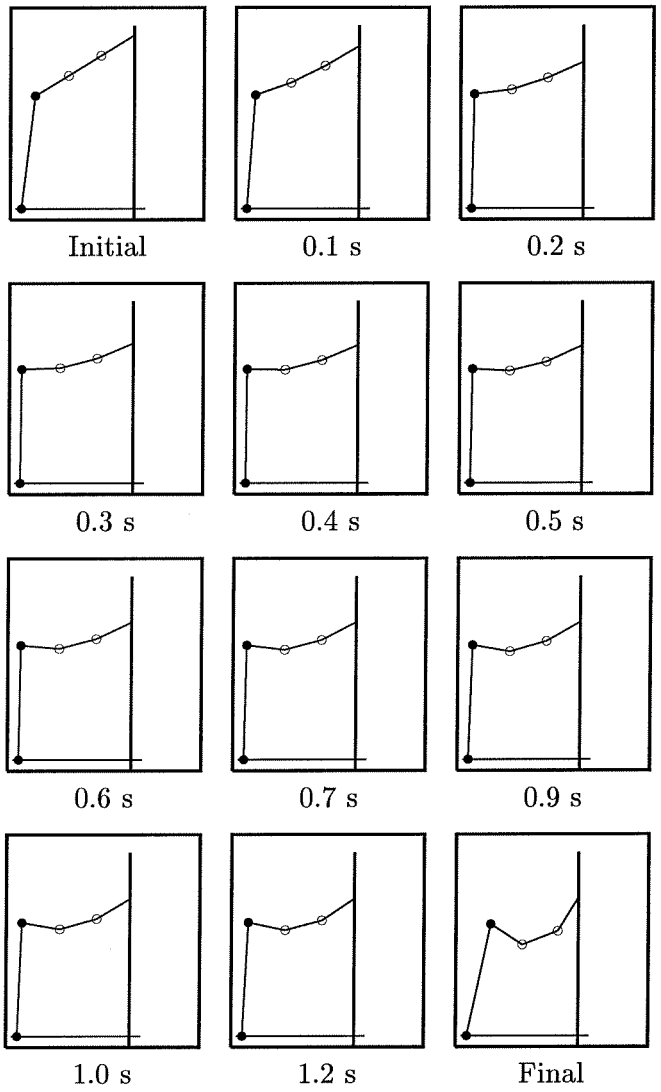


Figure 5.4 Instantaneous Jacobian controller ($m = 0.005$ kg, $k = 0.01$ N/rad).

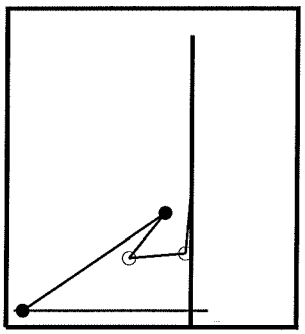


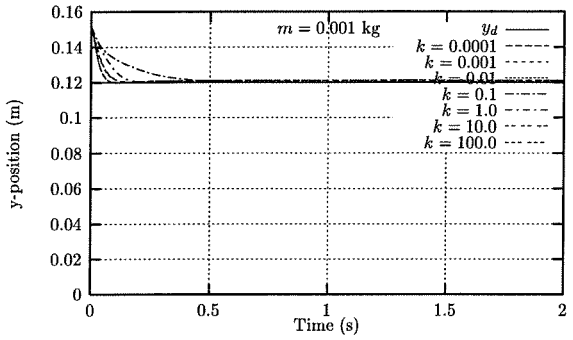
Figure 5.5 A "too flexible" manipulator ($k = 0.0001$ N/rad).

5.2.2 Effect of changing flexibility

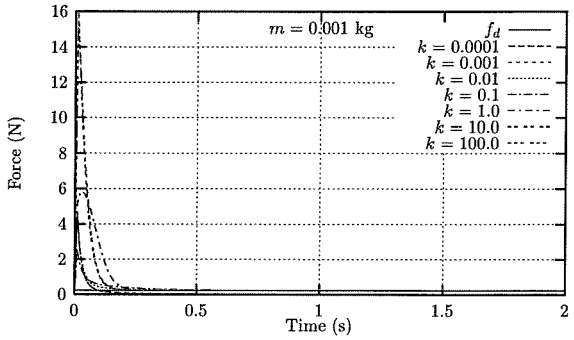
The flexibility of the simulated model is parameterized by the spring constants of the unactuated joints. As the spring constants are reduced the effective flexibility of the links decreases. The controller gains are set constant for all the simulations for each controller. In addition, the J_* and instantaneous Jacobian controllers have the same workspace gains. The joint PD controller behaves significantly differently from the other two, and therefore we do not present the data for that controller on the same axes as the other two (see Figure 5.6). Obviously the two most flexible links are too flexible and cannot apply the force required at the tip. Recall that the proof of stability of the controllers required the assumption that the beams were stiff enough to apply the required end-effector force. Figure 5.5 shows the final configuration of the manipulator with spring constant $k = 0.0001$ and the instantaneous Jacobian controller. It might seem from the graphs in Figure 5.6 that the joint PD controller manages to control both the position and the force of the two most flexible manipulators. This is not true. As the equilibrium could not be calculated for these we used the equilibrium data for the next stiffest manipulator. As the actuated joint angles of the manipulator are controlled to values which are correct for one possible solution to the kinematic problem, and the beam bends only in its first mode, the manipulator converges to the correct kinematic solution. The force regulation however does not converge to the correct value. Similarly to the other controllers the force is much lower than that commanded (and is obfuscated by the scale of the graph).

It is interesting to note that the behavior of the system controlled by the instantaneous Jacobian and the J_* controllers are very similar for all the flexibilities for which they could be stabilized. The joint PD controller shows a relatively larger variation in its behavior for differing flexibilities. Disregarding the two most flexible manipulators, the final position is achieved earlier with increasing stiffness. In the force response though the initial force excursion also increases with the stiffness. This exhibits the effect of stiffer links, and in the limit rigid links. The ability of the compliance to absorb the initial force excursion is what makes it attractive for contact tasks. The workspace controllers are able to greatly ameliorate the initial transient. Note that even in the case of instantaneous Jacobian controller,

the $k = 0.01$ N/rad manipulator shows force excursions (both above and below the commanded force) slightly smaller than the more stiff manipulators. The J_* and the instantaneous Jacobian controllers perform very similarly as far as position tracking is concerned when we have sufficient stiffness. Recall that they have identical gains. However, they show very different behavior in force regulation. The J_* controller achieves much better force regulation, with no initial overshoot in the force applied.

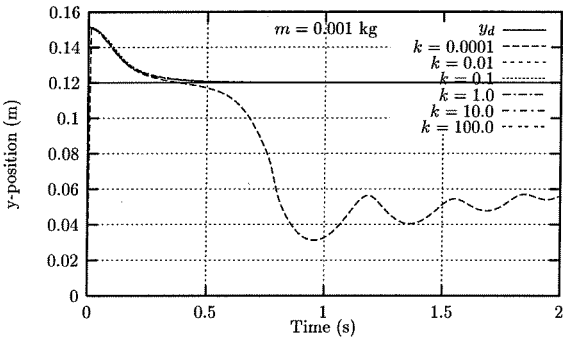


(a) Position tracking.

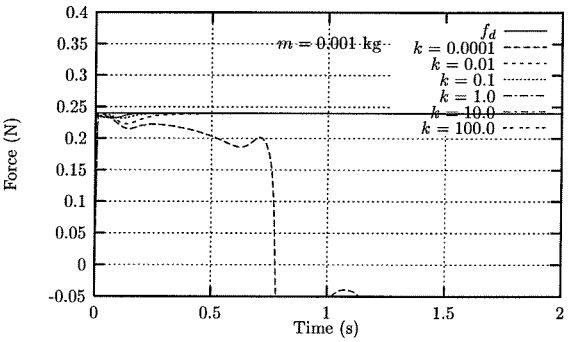


(b) Force regulation.

Joint PD

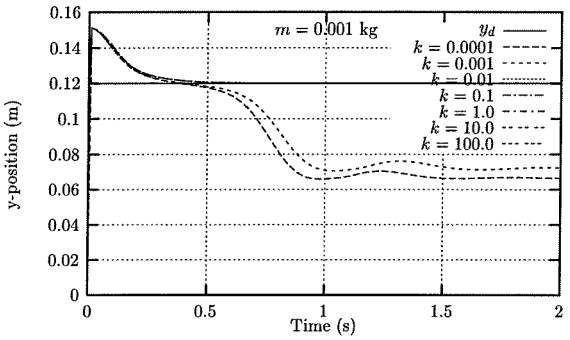


(c) Position tracking.

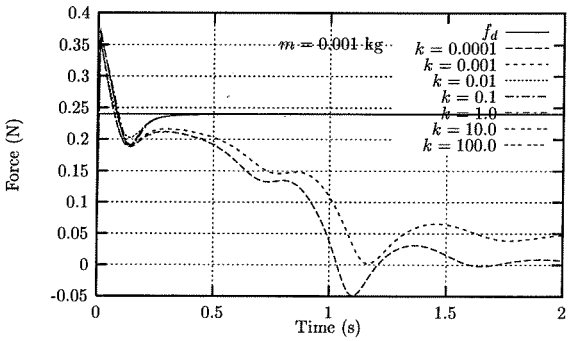


(d) Force regulation.

$\mathbf{J}_*$



(e) Position tracking.



(f) Force regulation.

Instantaneous Jacobian

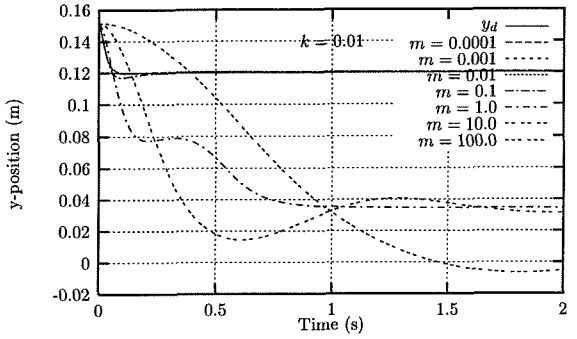
Figure 5.6 Effect of changing flexibility.

5.2.3 Effect of changing mass and damping

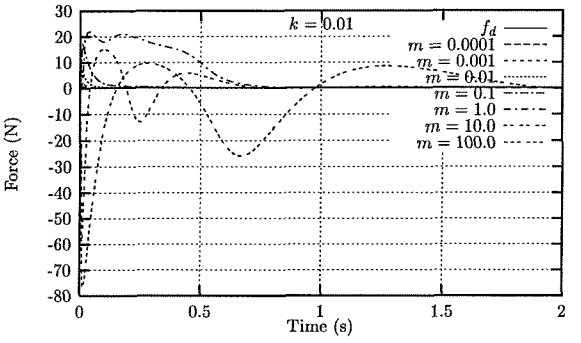
The effect of changing mass is illustrated in Figure 5.7. For these simulations the damping of the unactuated joints was varied as $\sqrt{m}$, in accordance with our singular perturbation analysis setup. The damping factor used was a constant for all the simulations and equal to $\eta = 0.1$. The damping at the unactuated joints is given by $\eta\sqrt{m}$.

The most interesting observation from the simulation is the operation of the boundary layer system. For the lighter manipulators the applied force very rapidly approaches that demanded and then stays close to it. The initial rapid approach to the commanded force is the boundary-layer system in action. As the mass is increased this effect decreases significantly and we see an increasingly gradual approach to the final force. Also of note is the fact that there is an initial negative force excursion for the heavier manipulators which is not present for the lighter manipulators.

The joint PD controller again behaves significantly differently than the workspace based controllers. It is unable to regulate position for the heavier manipulators while the other two controllers always manage to regulate position. Position tracking behavior of the J_* and the instantaneous Jacobian based controller are very similar. Both perform better with lighter links. Force regulation is different though, and the J_* controller achieves less overshoot in almost all the simulations.

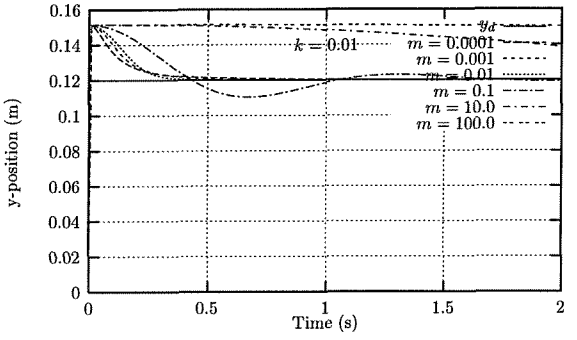


(a) Position tracking.

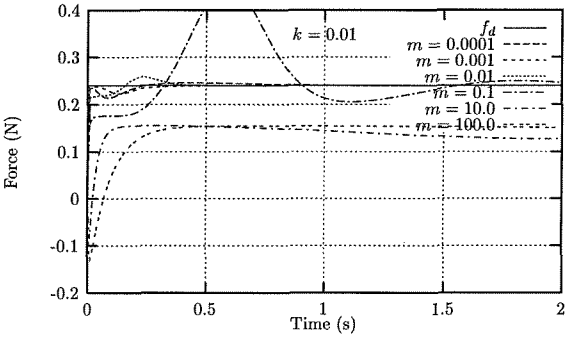


(b) Force regulation.

Joint PD

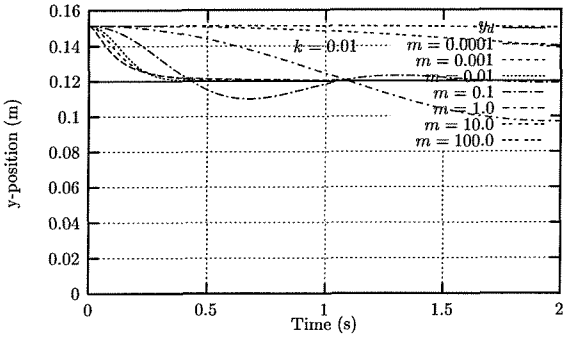


(c) Position tracking.

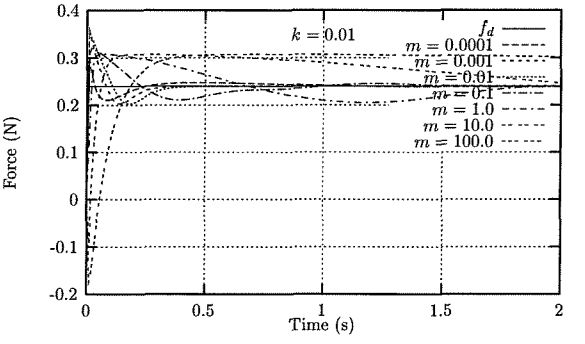


(d) Force regulation.

$\mathbf{J}_*$



(e) Position tracking.



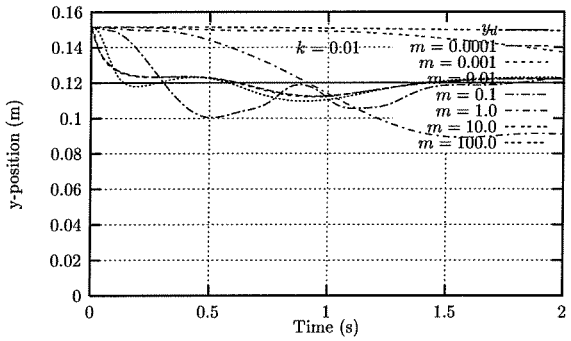
(f) Force regulation.

Instantaneous Jacobian

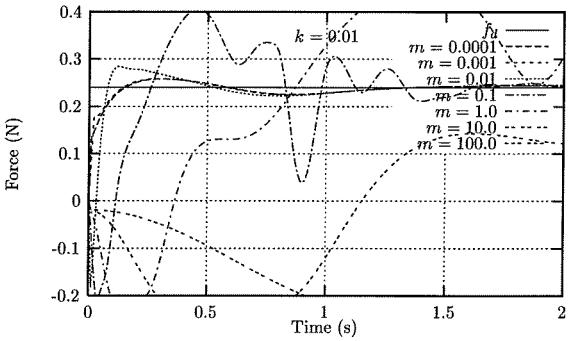
Figure 5.7 Effect of changing mass and damping.

5.2.4 Effect of changing mass keeping damping constant

In our analysis we proposed that damping of the unactuated joints (flexibility) be changed in proportion to the square-root of the mass. We performed simulations of the instantaneous Jacobian and the J_* based controllers in which we changed the mass of the sublinks keeping the damping of the unactuated joints a constant. This was done to test how crucial the assumption of the coupled mass and damping scaling was to the performance of the controllers. The results are shown in Figure 5.8 and 5.9. For the simulations in Figure 5.8 the damping ratio was made lower than the lowest damping ratio (for the lightest manipulator) in Figure 5.7. It is evident that the performance for the heavier manipulators deteriorates significantly. For a higher value of the damping ratio (Figure 5.9), equal to the damping ratio for the lightest manipulator in Figure 5.7 the performance is somewhat better. The performance however is still significantly worse than the performance for the manipulators with the damping factor scaled from the square-root of the mass. It is interesting to note that the deterioration in force regulation is more acute than that in position regulation. This shows that scaling the damping ratio with the mass in the manner of our analysis, does play an important part in the control of the manipulator. This reinforces the design guideline mentioned, for ease of controllability of a flexible manipulator at the end of Section 3.2.2.

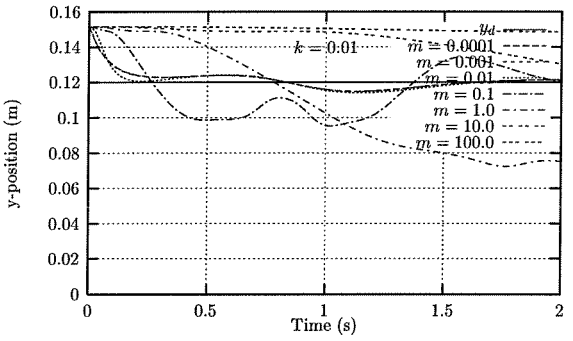


(a) Position tracking.

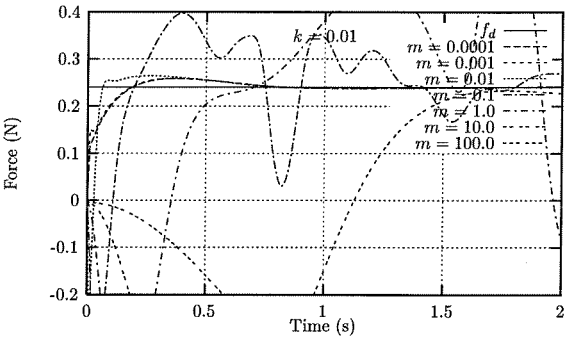


(b) Force regulation.

$\mathbf{J}_*$



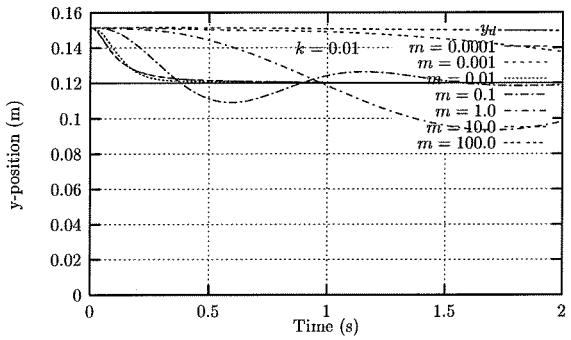
(c) Position tracking.



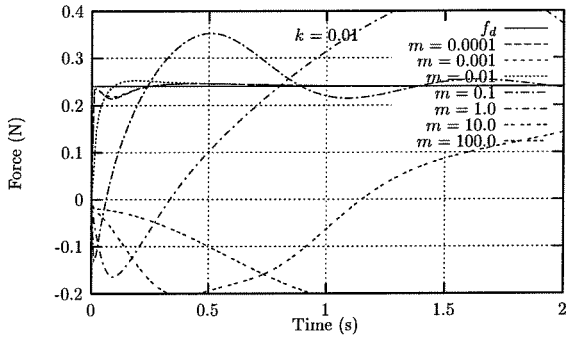
(d) Force regulation.

Instantaneous Jacobian

Figure 5.8 Effect of changing mass (damping coeff. = 0.0001 Ns/m).

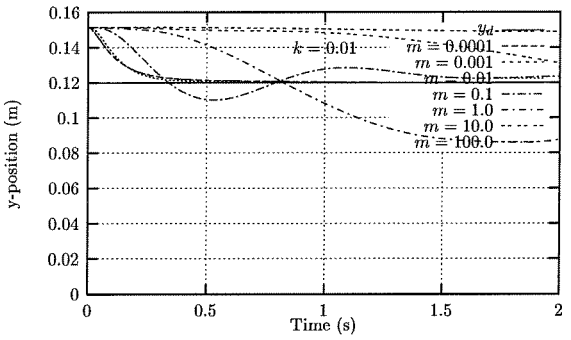


(a) Position tracking.

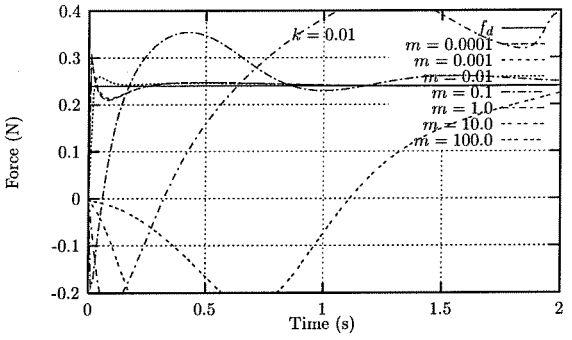


(b) Force regulation.

$\mathbf{J}_*$



(c) Position tracking.



(d) Force regulation.

Instantaneous Jacobian

Figure 5.9 Effect of changing mass (damping coeff. = 0.001 Ns/m).

5.3 Discussion of Simulations

The simulation results point at the significantly different character of the three controllers we proposed. The joint PD based controller showed the fastest convergence time, at the expense of a much greater force transient (by an order of magnitude). Further, it also showed poorer convergence properties for heavy manipulators. All the controllers were able to control the stiffer manipulators, and therefore by extension it would appear that they would all be able to control rigid manipulators. The J_* and the instantaneous Jacobian based controllers were able to achieve very similar performance over a wide range of flexibility. The joint PD based controller showed a more varied behavior. The J_* and instantaneous Jacobian controller were able to control heavier manipulators, beyond the assumption of lightness which was implicit in their analysis.

The J_* and the instantaneous Jacobian based controllers differ slightly but significantly in the control law they implement. The additional assumption required to be made to prove the stability of the instantaneous Jacobian control law is the insensitivity of the tip position to the change in the flexibility state variables (the Ψ's) in the reduced order system. The position regulation behavior of both the controllers is very similar. Figures 5.3 and 5.4 show that during the initial phase of motion under either control law the flexion of the beam is very small. However, once the tip of the beam is close to the final position the tip moves relatively little as the beam is flexed to its final shape. Therefore the assumption required for the stability of the instantaneous Jacobian based controller is largely satisfied. It is interesting to note the difference in the force regulation behavior of the two controllers. The better performance of the J_* controller is due to the incorporation of the flexibility model in its control action. In one sense, the J_* controller computes out the values of the flexibility state variables, almost like an observer, and uses that knowledge in its computation of the control action.

The variation of performance with the mass of the manipulator, keeping the damping of the system constant, showed the importance of the scaling procedure for the damping. The performance of the controllers deteriorated significantly when the damping ratio was not scaled. This reinforces the design guideline for ease of controllability of flexible manipulators mentioned at the end of Section 3.2.2.

Taken with the discussion presented at the end of the last chapter regarding issues of implementation of the the three controllers, the instantaneous Jacobian based controller seems to be the most attractive for experimental implementation. In the next chapter we present experimental data from an implementation of the instantaneous Jacobian based controller for grasping with flexible link robots.

Chapter 6

Experimental Evaluation

Analysis and simulations, no matter how detailed cannot fully model the realities of actual experiments. The uncertainties and complexities of real systems are very difficult to frame completely in the mathematics of the analysis and simulation. Therefore, we used our control methods in experiments to verify their validity on a real system. This chapter presents experimental results obtained using a planar, reconfigurable, two-finger robot hand, in grasping experiments. Rigid and flexible link robots with different flexibilities were used for experimentation.

The experimental testbed is described in Appendix A.

6.1 Experiments on Robotic Grasping

The theory developed for control of flexible robots was developed in the setting of a single robot manipulator in a constrained motion task. To show that the theory extends to the case of robot manipulation in general (and grasping in particular) we performed grasping experiments with rigid and flexible links on our reconfigurable robotic setup. Figure 6.1 is a picture of the robotic setup performing flexible grasping. The steel rule in the picture is 6 inches long. In this section we present the aims of our experimentation and details of the procedure used for its implementation.

6.1.1 Aims of experimental work

Our experimentation had two main aims.

Validation of control method: Though mathematical analysis and simulation are important tools for determining the viability of a control method, its practicality is best demonstrated by actual experimentation. Another very important conclusion that can be drawn qualitatively from experimentation is that of robustness of the control methodology. Robustness issues due to errors in modeling and unmodeled dynamics, which are very difficult to extract from mathematical and numerical treatments, especially for complex nonlinear systems, can be observed in experimentation.

Effect of flexibility: One of the aims of our work is to show the advantages of flexibility in certain manipulation tasks. This is most naturally exhibited through

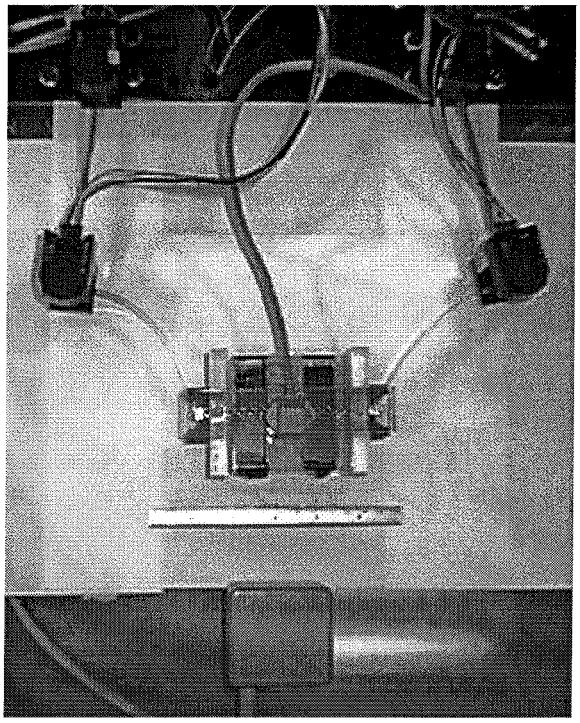


Figure 6.1 Grasping with flexible links.

experimentation, because the advantages we claim for flexibility are related to robustness of the force of manipulation in a hybrid force-position control scheme.

6.1.2 Experimental implementation of controllers.

A detailed description of the experimental setup is provided in Appendix A of this thesis. The experimental implementation of the controllers for the grasping experiments is shown in Figure 6.2. The workspace trajectory for the object is generated in software. The actual position of the object is determined from the Polhemus™ six degree of freedom position sensor. The force required for position correction in the workspace is calculated. The values of the forces depend on workspace control implemented. We implemented a simple workspace PD controller to determine the position correction forces in the workspace for the experimental data reported. Any other control method would work as well. We did try a LQR controller but were unable to get much performance with the implementation because of difficulty in modeling the experimental setup accurately. The internal force specified is added to the corrective force to make up the total workspace force required.

The corrective joint torques required are calculated from the workspace forces by using the workspace Jacobian of each finger. The Jacobian is a kinematic calculation and depends on the joint angles and the link lengths of the fingers. In case of the rigid robot, the link lengths are constants and the joint angles are directly

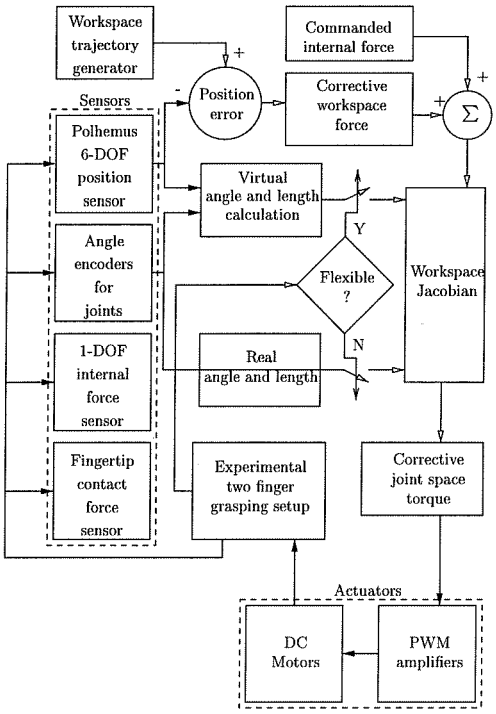


Figure 6.2 Experimental implementation of controllers

obtained from the angle encoders at the joints. For the flexible robots the "virtual" lengths and angles of the flexible link are calculated from the information about the joints and endpoints of the robot. In our setup this information was obtained from the angle encoders and the object position sensor. Note that in general it is not required that we have an object position sesnsor. We only used the knowledge of the object position to calculate the positions of the finger tips. In an application where arbitrary objects are to be manipulated, the tip sensors would be at the tips of each finger. The routines for the calculation of the Jacobian are provided with the actual link lengths and angles for the rigid robot, while the virtual lengths and angles are provided for the flexible calculation. Note that the Jacobian calculation is done by the same routine for both cases.

The joint space torques are translated to motor currents through PWM amplifiers. Due to the tendon drive the two motor torques for each finger are not independent, and we have to use a transformation matrix to calculate the motor torques from the joint torques. This extra transformation would not be required for direct drive robots.

The computational platform used was a IBM-PC compatible, with an Intel-80486 processor, running at 66 MHz. The real-time software used for implementation was Sparrow-2.1 [39]. Sensor data is collected at 150 Hz. and filtered digitally in software. The controller was operated at 50 Hz. The bandwidth of each finger was measured to be of the order of 5 Hz. The computation times required for the

Controller	Execution time (ms)
Workspace PD (Rigid)	1.320
Workspace PD with Instantaneous Jacobian (Flexible)	1.392

Table 6.1 Execution time.

Finger Bases (cm)		Circle Parameters (cm)		
Finger 1	Finger 2	Center	Diameter	Frequency(Hz)
$x: -10.97$	$x: 10.84$	$x: 0.0$	3.0	$\frac{1}{6}$
$y: -10.60$	$y: -10.31$	$y: 8.5$		

Table 6.2 Parameters for tracking experiments.

controllers is given in Table 6.1.

6.1.3 Description of experiments

In this section we provide an overview of the experimental procedure followed to gather data. As the aim of the experiments includes comparisons of performance of different setups, the experimental procedure is itself of importance in interpretation of the data.

Data was collected while the grasping setup described earlier in this chapter was trying to track a circle. The particulars of the setup and the circular path are presented in Table 6.2. For each experiment, data capture was turned on with the grasping setup tracking the object and regulating the internal force. For the first twelve seconds of data capture (two traverses of the circular path) both the position tracking and the force regulation were kept on (subcycle-A). At the end of the twelve seconds the position tracking was switched off, however the force regulation was maintained for a further two seconds (subcycle-B). At the end of the additional two seconds (fourteen seconds from the beginning of the data capture) the force regulation was turned off too and data was captured for an additional two seconds (subcycle-C). A graphical representation of the data cycle is presented in Figure 6.3.

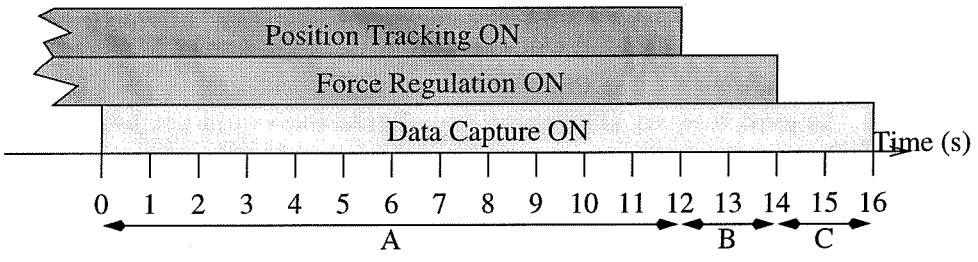


Figure 6.3 Data cycle for tracking experiment.

Subcycle-C of the data cycle was used to calculate the "zero"-force for that data run. This was required because of the shift in the zero force position of the force sensor being used between data runs. Subcycle-B was used to find out the internal force (F_0) in the absence of any position correction forces. This was used as the correct internal force demanded during the data run. Note that due to the presence of significant friction and stiction in the tendon drives and between the object and the supporting plate and due to the flexibility of the tendons themselves the actual internal force achieved was not identical to the desired internal force. Further, the force transducer only detected part of the internal force due to the construction of the object. Subcycle-A was then used to compute both the error in tracking position and the errors in regulating the internal force. A typical data cycle is presented in Figure 6.4.

During experimentation, the parameters of the circle to be tracked were kept constant. The internal force desired was changed between experimental runs. The actual internal force achieved by the grasping setup varied from run to run due to the nature of the experimental setup. This was however accounted for as described in the foregoing paragraph. At each setting of the internal force, data was taken for multiple sets of gains each of which provided reasonable performance. We present the effect of change in gain in our section on presentation of data. Note that only the proportional gain was changed. The derivative gain was related to the proportional gain by a time-constant, which was kept at a constant of 0.02 $\frac{1}{s}$ throughout the experimentation.

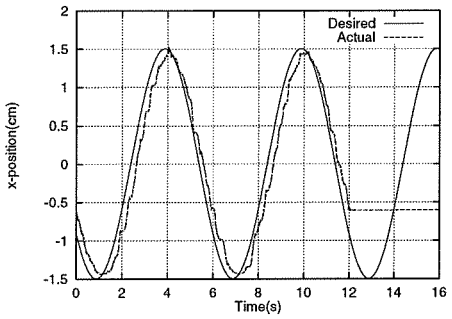
During experimentation, it was observed that there was significant stiction between the object being grasped and the lucite base plate on which the experimental setup is constructed (refer to Appendix A). To ameliorate these effects we used a Teflon sheet between the lucite plate and the object. Graphite powder was also used as a dry lubricant on the Teflon sheet. Though this resulted in noticeable improvements, the effects of friction and stiction were still significant. However, as all the controllers had to surmount this and modeling of the effect was very difficult, the problem was treated as an unmodeled behavior of the system that was one test of controller robustness.

6.2 Performance Measurement from Experimental Data

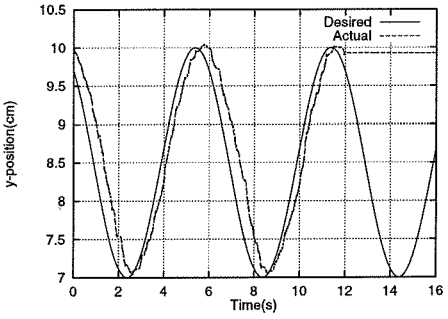
To determine performance a few performance indices were defined. Comparison of controller/flexibility combinations was treated in the framework of these performance indices. We present the performance indices used in what follows.

Root-mean-square error in position (P_{e_2}): This is the averaged 2-norm of the positional error during subcycle-A. This was used as a measure of position tracking performance. A lower P_{e_2} denotes better performance in position tracking.

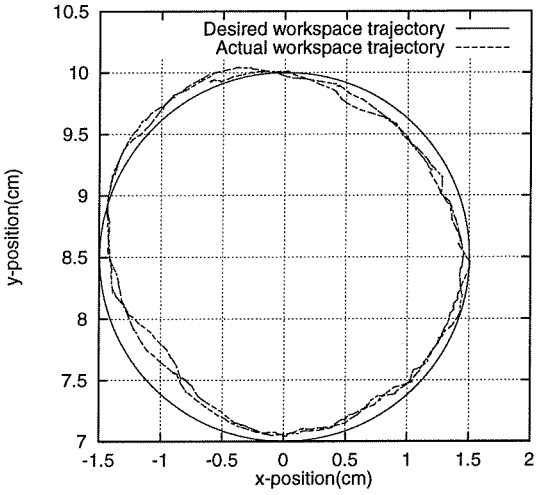
Averaged high frequency 2-norm of position (P_{hf_2}): This is the averaged 2-norm of the high frequency component of the position signals. The actual method of computation is to take the average 2-norm of the difference of the original x and y position signals and their low-pass zero-phase forward and reverse digital filtered



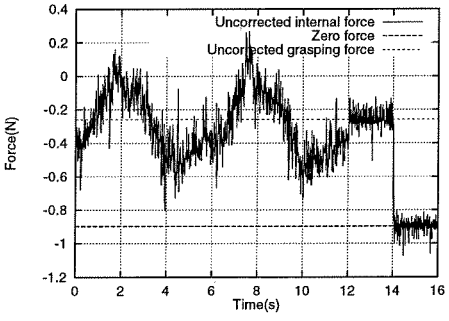
(a) Time trace of x-position.



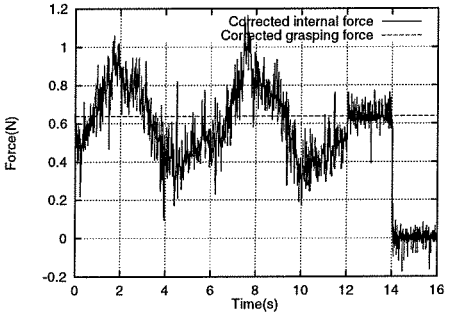
(b) Time trace of y-position.



(c) Workspace trajectory.



(d) Uncorrected force.



(e) Corrected force.

Figure 6.4 Data from experimental run.

versions. The filtering is done using the `filtfilt` command in `MATLAB` and results in zero phase distortion of the signal. The filter used was a 5th order Butterworth filter, with cutoff at 7.5 Hz. A lower P_{hf_2} denotes more steady tracking.

Root-mean-square error in internal force (F_{e_2}): This is the averaged 2-norm of the error in internal force during subcycle-A. The error in internal force is defined as the difference between the force during subcycle-A and the average force during subcycle-B (after discarding datapoints at either end of subcycle-B). A lower F_{e_2} should denote a better performance in force regulation.

Averaged high frequency 2-norm of force (F_{hf_2}): This is the averaged 2-norm of the high frequency component of the force signal. The actual method of computation is to take the average 2-norm of the difference of the original force signal and a low-pass zero-phase forward and reverse digital filtered version of the force signal. The filtering is done using the `filtfilt` command in `MATLAB` and results in zero phase distortion of the signal. The filter used was a 5th order Butterworth filter, with cutoff at 7.5 Hz. A lower F_{hf_2} denotes less fluctuation in the applied force.

Averaged 2-norm of negative force (F_{-2}): This is the average (over the data subcycle A) of the 2-norm of the force below zero applied by the fingers. A F_{-2} of zero denotes that the fingers were able to maintain a positive internal force on the object through out the motion. Any value above zero indicates that there were times when the fingers were not able to apply any internal force on the object and if unrestrained would have dropped the object.

Minimum applied force (F_{min}): This is the minimum force applied during subcycle A of the data run. A negative value denotes that the internal force was relaxed completely and the magnitude of the negative force points out the extent of the error. Conversely, a positive value denotes that internal force was maintained and its magnitude is a measure of the minimum robustness of the grasp.

Average proportional gain (G_{kp}): This is the average of the workspace proportional error gain used during the experiment. The corrective force applied during the experiment is proportional to the G_{kp} as the derivative gain is scaled from this by a time-constant. We used a constant time-constant throughout all the experiments.

Average tracking force (F_{tr}): This is the average tracking force applied during the experiment. It is the product of P_{e_2} and G_{kp}. It gives an idea of the force required solely for the tracking part of the control. As real systems have actuator limitations it is important that the tracking force be as low as possible to avoid saturation of the actuators.

For ease of referral the indices defined above and some other quantities we use in presenting experimental data, their symbols and units are presented in Table 6.3. In experimental work dimensionless numbers are often sought to characterize data such that it may hold for systems other than the experiment itself. We found the dimensionless numbers given in Table 6.4 useful for characterizing our data.

Index description	Symbol	Units
RMS position error	P_{e_2}	cm
Averaged high frequency 2-norm of position	P_{hf_2}	cm
RMS force error	F_{e_2}	N
Averaged high frequency 2-norm of force	F_{hf_2}	N
Averaged 2-norm of negative force	F_{-2}	N
Minimum force	F_{min}	N
Maximum force	F_{max}	N
Force range ($F_{max} - F_{min}$)	F_{ran}	N
Internal force	F_0	N
Average tracking force	F_{tr}	N
Average proportional gain	G_{kp}	N/cm

Table 6.3 Defined quantities for experimental results.

Name	Description	Symbol
Normalized RMS position error	$\frac{P_{e_2} - \min P_{e_2}}{\max P_{e_2} - \min P_{e_2}}$	$\overline{P_{e_2}}$
Normalized averaged high frequency 2-norm of position	$\frac{P_{hf_2} - \min P_{hf_2}}{\max P_{hf_2} - \min P_{hf_2}}$	$\overline{P_{hf_2}}$
Normalized RMS force error	$\frac{F_{e_2}}{F_0}$	$\overline{F_{e_2}}$
Normalized averaged high frequency 2-norm of force	$\frac{F_{hf_2}}{F_0}$	$\overline{F_{hf_2}}$
Normalized averaged 2-norm of negative force	$\frac{F_{-2}}{F_0}$	$\overline{F_{-2}}$
Normalized minimum force	$\frac{F_{min}}{F_0}$	$\overline{F_{min}}$
Normalized force range	$\frac{F_{ran}}{F_0}$	$\overline{F_{ran}}$
Normalized tracking force	$\frac{F_{tr}}{F_0}$	$\overline{F_{tr}}$

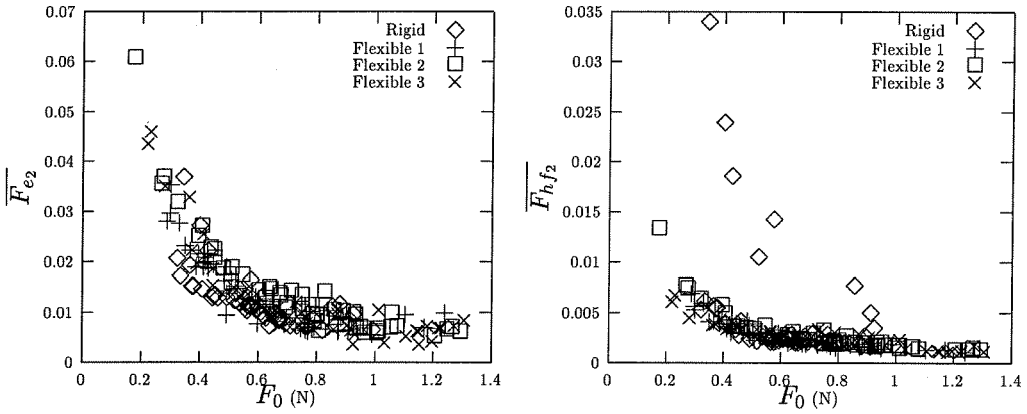
Table 6.4 Dimensionless numbers for experimental results.

Link set	Thickness (cm)	Length (cm)
Rigid	0.1651	11.42
Flexible 1	0.0508	11.56
Flexible 2	0.0381	11.53
Flexible 3	0.0254	11.53

Table 6.5 Link sets used in experimentation.

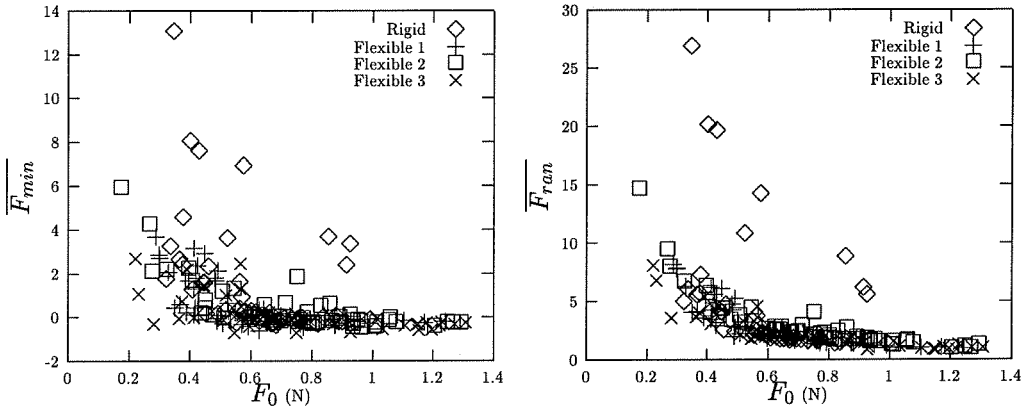
6.3 Experimental Results and Discussion

In this section we present our experimental results and discuss them within the framework of the performance indices mentioned in the previous section. We performed experiments with four different sets of links with different flexure. Data was collected for different internal forces and multiple runs were made for each internal force, link set combination. The particulars of the link sets are presented in Table 6.5, in order of increasing flexibility. The last link is extremely flexible and during grasping we could bend it much beyond the linear range. None of the flexible link robots (including Flexible 1) could be controlled by the rigid robot controller, at any setting of the internal force.



(a) $\overline{F_{e_2}}$ (b) $\overline{F_{hf_2}}$

Figure 6.5 Effect of internal force on relative force errors.



(a) $\overline{F_{min}}$ (b) $\overline{F_{ran}}$

Figure 6.6 Effect of internal force on relative force errors.

6.3.1 Overall trends in tracking data

We first present the full data for all the link sets together to determine the overall trends. It is important to keep in mind the conditions under which the experiments were done. There was significant friction and stiction in the tendon drives as well as between the object and the base plates. The stiction was specially bothersome for the flexible links as they tended to excite the flexible modes.

The effects of increasing internal force are presented in Figures 6.5 and 6.6. As can be seen, the errors in force, as measured by any of the performance indices, relative to the internal force being applied decrease significantly as the internal force is increased. However, the trend is very well behaved in case of the flexible fingers. The rigid fingers do as good a job of regulation of force in terms of the absolute error (as measured by F_{e_2}) but can do significantly worse in all the other measures. This means that though they might on the average control the force as well as the flexible robots, the contact forces they apply can be very erratic, thereby leading to the chance of dropping the object.

Figure 6.7 is a comparison of the two position performance indices. The normalized dimensionless versions are used, to make the comparison more meaningful. It is interesting to note that the best performances in both the indices were attained by the rigid robot (a zero in either index indicates the best performance, not a perfect performance). As is evident, good performance in one index leads to worse performance in the other. Thus the question of better position tracking performance can only be answered in an application specific context. There is an area in the middle of the graph which we can consider acceptable performance in terms of both indices. The rigid fingers can provide much better than acceptable performance in the two norm error at the expense of being more erratic (increased jittery tracking). Notice that the flexible fingers tend to clump around the region of acceptable performance

in both and exhibit performance very similar to the rigid fingers in the region in which they work. Indeed, from this figure it is evident that depending on what exactly our specification of position performance is we can get a different answer to the question of the better setup for the task.

The proportional gain in the workspace PD control provides the workspace position error correction force. In our control laws (as mentioned previously) the proportional gain was used to scale the derivative gain too, via a time-constant which was kept constant for all the data collected. The graphs of the effect of changing gain (Figure 6.8) show that P_{e_2} performance gets better with increasing gain, as expected. Further, as the gain was increased the flexible fingers achieved better P_{e_2} performance relative to the rigid fingers at the same value of the gain. Of course, the rigid fingers were able to achieve a much better overall P_{e_2} performance with very high gains. The P_{hf_2} performance is degraded for all the link sets with increasing gain. At low gains the rigid fingers outperform the flexible. The F_{hf_2} force performance shows a slight worsening with increasing gain. However, at high gains the rigid fingers can do extremely poorly indeed. In actual experimentation the flexible fingers could not be operated at very high gains. The maximum gain which could be applied to the flexible link fingers increased with increase in stiffness of the links.

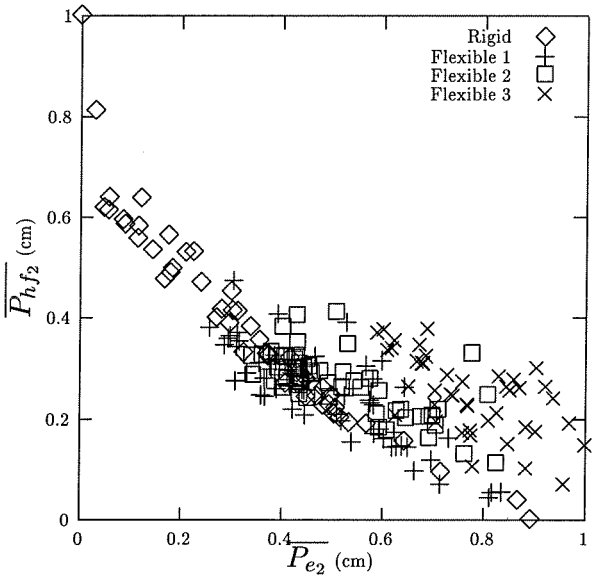
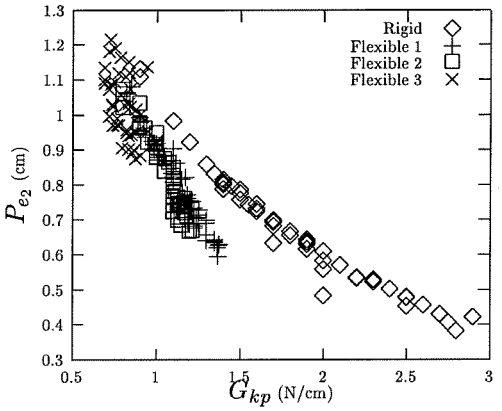


Figure 6.7 Comparison of position performance indices.

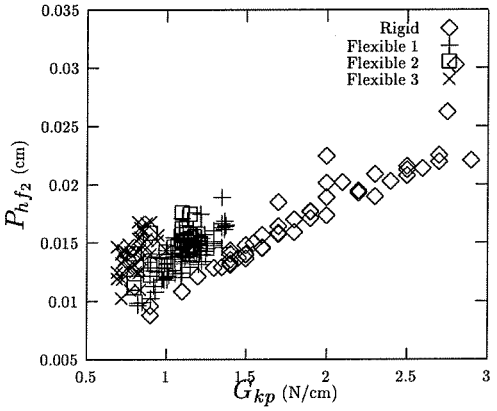
Another perspective of the overall performance of the grasping setup is obtained by considering the position error correction force. The separation between the rigid and the flexible fingers is very apparent in Figure 6.9. A lower F_{tr} is preferable because in real systems we always have actuator saturations. From the graphs it is quite evident that the rigid robot consistently required more position correction force than the flexible. Note that this did not always translate to better position control

performance for the rigid robots (Figure 6.10), especially in terms of $\overline{P_{hf_2}}$. Also interesting to note is the somewhat layered appearance of the graph in Figure 6.9, in that the more flexible links appear at lower positions in the graph. Partly, this reflects the lower gains used for the flexible links. However, the performance in position tracking also matters.

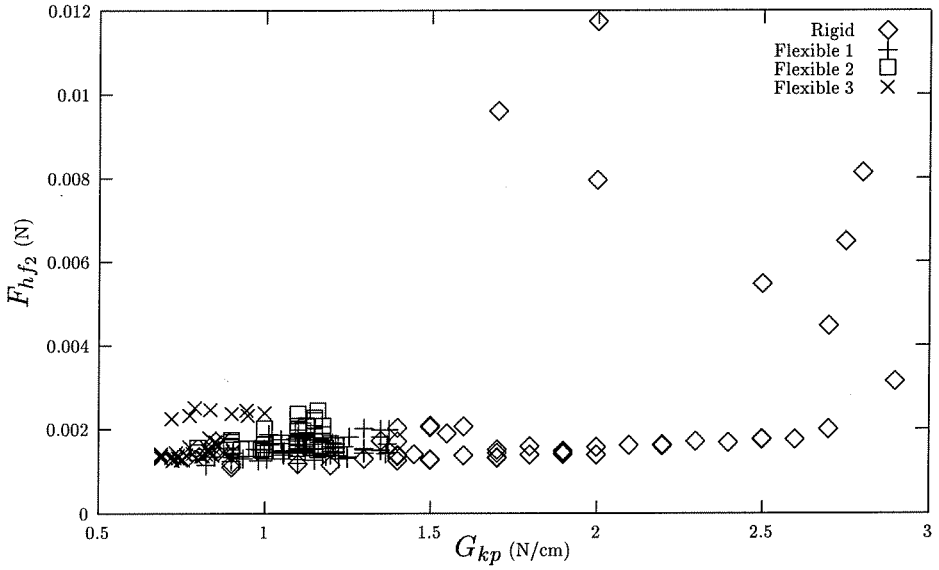
We shall discuss these results further at the end of this chapter.



(a) P_{e_2} performance.

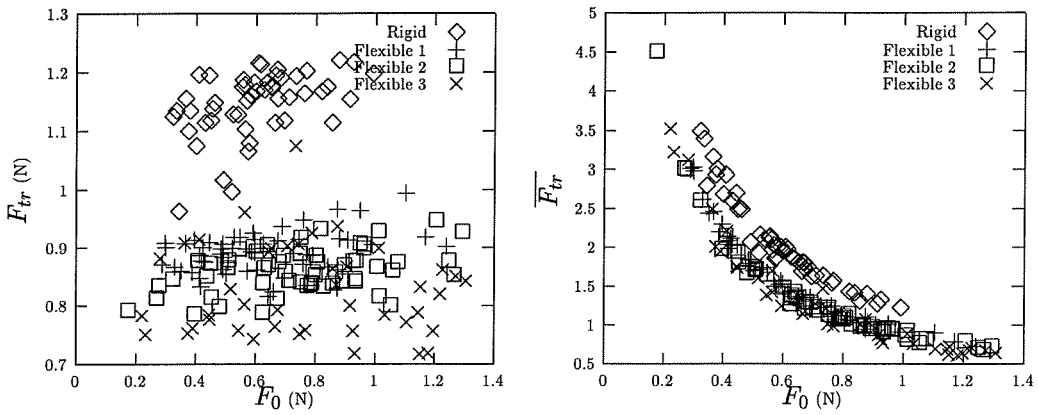


(b) P_{hf_2} performance.



(c) Force performance.

Figure 6.8 Effect of gain on performance.



(a) Error correction force.

(b) Normalized error correction force.

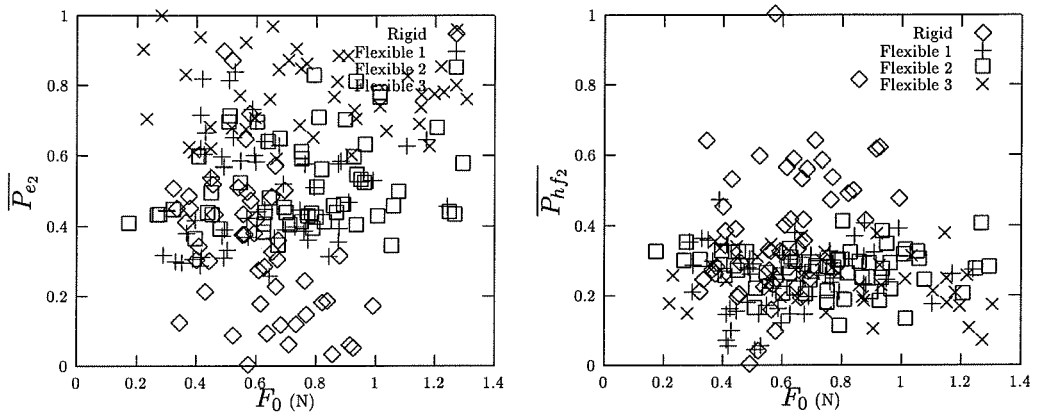
Figure 6.9 Error correction force.

6.3.2 Experimental results for a single link set

In addition to the comparative presentation of data, it is interesting to look at the behavior of flexible links in hybrid force/position control tasks. In this section we present results for the link set "Flexible 2" (refer to Table 6.5), during one data cycle.

Figure 6.11 and 6.12 show the behavior of the flexible beam during an experimental run. Data is presented at one second intervals. The data used for the calculations was the original, unflexed beam length and the virtual length, which was calculated from the sensed data of the object position and orientation. It was assumed that the beam bent symmetrically in a circular arc. In actual practice this is not true and the beam bends much more at its base than at its tip. The lightly shaded bent link in Figure 6.11a exhibits the case when the base of the beam bends more (smaller radius of curvature) than the tip, for the same virtual length. Therefore the data for deflection presented here must be considered conservative.

It is observed that the virtual length of the link does not change by very much and is close to the original length of the unflexed link. However the resultant deflection of the tip suffered by the flexible link is substantial, up to 90% of its length. The linear beam theory does not apply in this regime. The amount of the deflection makes clear the reason why a "rigid controller" will not suffice. The rigid controller would work under the assumption that the tip of the flexible link was at the tip of the unflexed link, which in reality can be very far away from the actual tip. In our experiments, the rigid controller was not able to control even the "Flexible 1" link set: the stiffest of the flexible link sets. Therefore, to be able to control flexible links of the order of flexibility we have used and experimented with, the controller has to consider the flexibility in its control action. The effect of the flexibility is severe



(a) $\overline{P_{e_2}}$. (b) $\overline{P_{hf_2}}$.

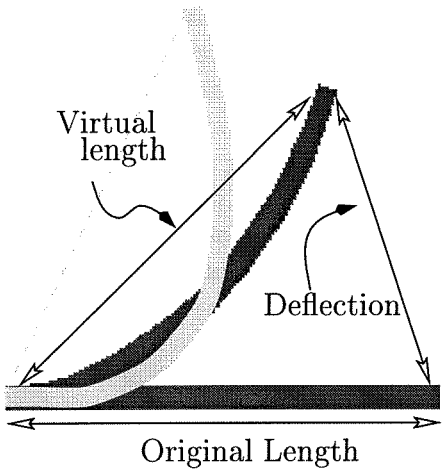
Figure 6.10 Effect of internal force on position performance.

enough to render it uncontrollable using the rigid controller. Figure 6.12 depicts the approximate shape and the orientation of one of the flexible links during the controlled part of the data cycle at one second intervals.

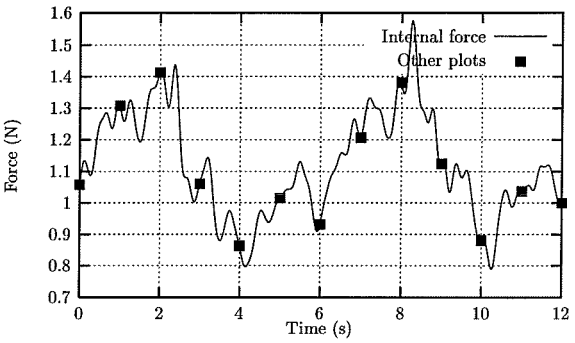
6.4 The Case for Flexibility

We have demonstrated through experiments that significant structural flexibility in manipulators can be controlled. The control methodology based on our analysis of a flexible manipulator pushing against a wall and described in previous chapters is able to perform satisfactorily in grasping situations as well. The flexibilities considered and experimented with are significant, and beyond the ability of the rigid controller to control. The additional requirements for the flexible controller to work are a sensor for the object position (or finger tip positions) and a marginally increased computation time on typical computation hardware.

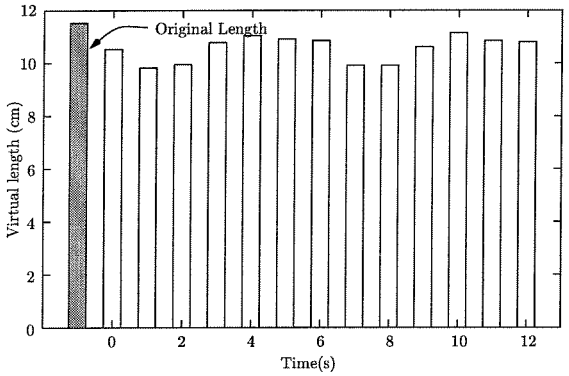
Analysis of experimental data shows that rigid link robots can perform better than flexible ones in reducing absolute error in position. However, this is achieved only at high position correction forces. At these high position correction gains the performance of rigid link robots is not very robust and there is degradation in internal force regulation. At comparable values of position control gains flexible link robots perform better than rigid link robots in control of absolute position errors. With increase in the internal force there is a strong trend of improvement in force regulation performance relative to the internal force. Lowering of the internal force can cause force performance of rigid link robots to become unpredictable, in terms of any of the performance measures. At any value of the internal force the rigid robot outperforms the flexible in terms of the absolute position error, but can show variable relative performance. The flexible link robots perform better in the high



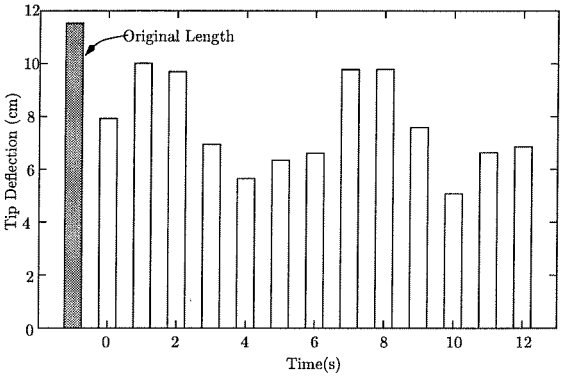
(a) Bending terminology.



(b) Force history.



(c) Virtual length.



(d) Beam deflection.

Figure 6.11 Behavior of "Flexible 2" link set.

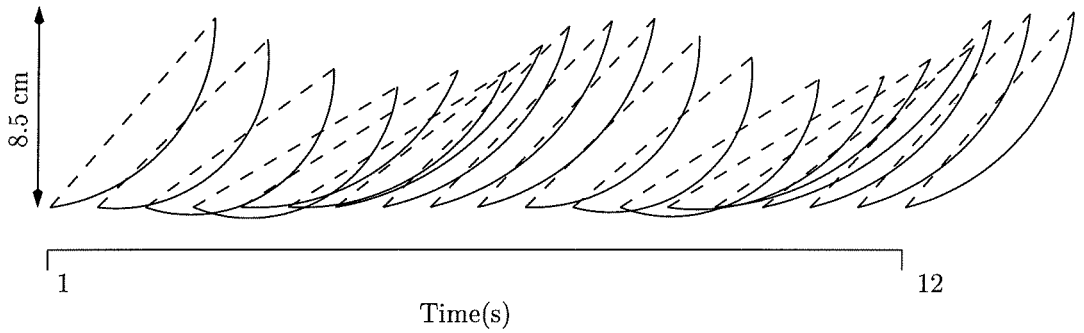


Figure 6.12 Bending of the "Flexible 2" link set.

frequency measure at almost all internal forces.

In light of the above, flexibility seems to be most advantageous in low internal force grasping tasks, in which rigid link robots show unpredictable behavior. In general flexible link robots show more robust behavior and follow trends more reliably. Actuator saturation is another scenario where the use of flexible link robots is attractive, as they achieve comparable absolute position tracking and better high frequency position tracking with lower actuation effort.

Coupled with the advantages mentioned in the introductory chapter of this thesis, the above makes a strong case for using flexible link robots in manipulation tasks, especially, as control of these robots in hybrid force/position tasks can be achieved with relatively simple, non computationally intensive modifications to existing workspace control laws.

Chapter 7

Conclusions and Future Work

We have tried to put forward a point of view about doing robotics using flexible manipulators. We believe that manipulators with flexible links can not only be controlled for hybrid force/position manipulation tasks, but can provide performance comparable to, and in some cases better than that of rigid manipulators.

7.1 Summary of Work Done

In this thesis, we have dealt with the problem of manipulation with flexible robots at different levels. We have analyzed the dynamics of single flexible manipulators in constrained motion tasks, mathematically, and have been able to set up a framework which can address significant flexibility in robotic manipulators. As a result of this analysis we were able to propose control laws which were provably stable. The analysis also provided a guideline for design of flexible manipulators to make control of these manipulators easier.

Simulations have been performed to test the performance of the controllers proposed and to determine the behavior of flexible manipulators in constrained motion tasks. In addition we have tested the effect on the flexible manipulator system, of the assumptions made during analysis.

The third part of the work presented was experimentation with rigid and flexible fingers in grasping tasks. The control laws used were developed during the analysis for individual manipulators in constrained motion tasks. One of the goals of the grasping experiments was to exhibit the applicability of the results of our analysis to an advanced hybrid force/position control task. The other major goal of the grasping experiments was to determine the tradeoffs inherent in the use of flexible manipulators as against rigid ones. This necessitated the development of a framework for measuring the performance of manipulators in constrained motion tasks. The analysis of experimental data within this framework provided insights into the tradeoffs involved in using flexible manipulators.

In summary we are able to state the following:

- Singular perturbation tools can be used to analyze significantly flexible manipulators in constrained motion tasks.

- Provably stable control laws for control of flexible robots in constrained motion tasks exist, and can be used to control robots with significant link flexibility.
- These control laws and their modifications can be implemented using existing technology on real setups.
- The modified Jacobians introduced can be used for mapping workspace control forces for flexible robots in the same way that the usual Jacobian is used for rigid robots. The workspace control law, which determines the workspace control force to be applied, can remain unchanged. Therefore the applicability of this work is quite general.
- Guidelines followed during robot design can make control easier. Scaling the damping coefficient of the flexibility of a flexible robot by the square-root of the mass of the flexible link is one such guideline. Material properties like density and viscoelasticity, as well as more esoteric strategies utilizing new materials like shape-memory alloys can be used to achieve the design.
- The extra effort of control of flexible manipulators can be offset by their enhanced performance in certain types of robotic tasks. Though rigid link robots can achieve better absolute tracking performance, flexible link robots exhibit better robustness properties in grasping tasks and their performance shows more reliable trends with changes in operating parameters. Flexible robots can achieve absolute tracking performance comparable to rigid robots and better high-frequency performance with lower actuation effort. Therefore, they are most suited for force/position tasks which require low forces to be applied steadily during motion.

7.2 Future Work

There are many avenues of fruitful research which can be followed for the control of flexible manipulators for robotic manipulation. We have been unable to prove the stability of the instantaneous Jacobian control law without making extra assumptions. However, this law works very well in practice. It would be interesting to investigate the properties of the system which ensure the success of this law. The insight gained would not only delineate the limits of applicability of this law but could potentially reveal more about the behavior of flexible manipulator systems. Design guidelines arising from such investigation would be a gratifying addition to current knowledge. Proof of asymptotic stability of the J_* controller is another avenue which could be explored.

Further experimentation geared towards implementing the J_* control law would also be welcome. We were unable to implement it due to the lack of dependable force sensors for the fingertips of our robot fingers. The basic control laws used during the experimentation reported in this thesis were quite simple. It would be interesting to experiment with more sophisticated control laws which push the performance of

the experimental setup to higher levels. The performance trends arising from such experimentation would be clearer than those obtained.

The picture which emerges from the experimentation carried out is still not very clear though certain trends are evident. More experimentation needs to be done to determine the behavior of flexible manipulators in constrained motion tasks. The framework for determination of performance should also be considered a first cut. As properties of such systems are determined, a standardised framework should evolve for performance measurement.

In our experimentation we used links of different flexibilities. The links were stainless steel feeler gages of different thicknesses and were purchased off-the-shelf. It would be an interesting exercise to explore using different materials, cross-sections and other design parameters, including the use of smart materials, the extent to which the scaling law proposed in our singular perturbation method holds in practice. It is worthwhile to know the additional design effort required to conform to the scaling law.

The general field of flexible robotics must be considered relatively young, and a lot remains to be done. The issues dealt with in this thesis are basic to a better understanding of the field and the problems that need to be addressed, as well as a justification for continuing work. The need for the future is diverse. More powerful analytical tools, a better understanding of the behavior of constrained flexible manipulators, improvements in sensing and actuation technology are all required before practical systems can be designed and built for real world applications.

Appendix A

A Reconfigurable Multi-Robot Testbed

The experimentation reported in this thesis was performed on a workbench designed and fabricated during the course of this work. The setup allows the use of multiple robots and is intended to be a testbed for performing various types of experiments in robotics. Each robot is a tendon driven, two degree of freedom manipulator. Reconfiguration of the construction of individual robots, as well as reconfiguration of robots in a multirobot setup are easily undertaken. Actuation is via DC motors driven by PWM amplifiers. Sensing of joint angles is through optical encoders. A six degree of freedom position and orientation sensor is available for sensing grasped objects. Strain and force measurement hardware and software is available as well. The whole setup is interfaced to a IBM-PC compatible for sensing, control and data-acquisition through commercial as well as custom designed interface hardware. Different control laws can be easily implemented on the setup by attaching controllers to the existing software base. This testbed has been used in the past for flexible-link robotics, flexible-actuation robotics, rigid and flexible grasping and for experiments in human-robot interaction.

A.1 Introduction and Overview

The motivation for building the robotic testbed was to provide a small-scale robotic setup, which allowed the possibility of reconfiguration for different types of experimentation, without large downtime or refabrication. The testbed went through multiple design cycles and evolved in both concept and design from the initial design goals.

One of the major design goals was *scalability*. We required the setup to be scalable to very small dimensions. This was due to a perceived potential application in medical/surgical endoscopy, where the space available for a finger like device is limited to the lumen of the endoscope, approximately a 3 mm diameter (please refer to Chapter 1). To ensure scalability the initial versions of the fingers had no non scalable parts. There were no metallic bearings at the joints. The material used to build the robot was delrin which has the frictional properties of teflon with better machining properties. This allows for finer tolerances in the actual building of the device. In later modifications we did introduce metallic bearings due to the stiction

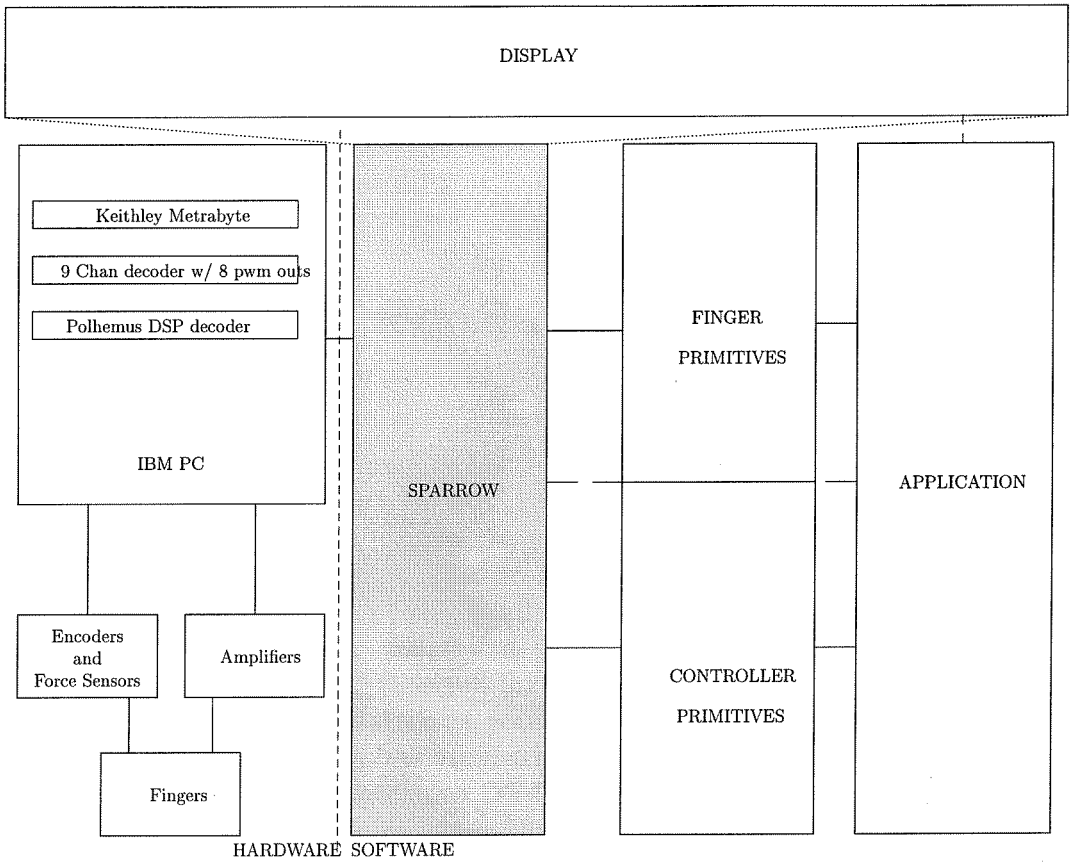


Figure A.1 Overview of experimental setup.

at the joints. Another of the main design goals was *reconfigurability*, to allow more than one set of experiments to be performed simultaneously. To achieve this the manipulators are modular, and a new manipulator can be pieced together from the parts in a very short time. Parts of the manipulators can be exchanged for other parts, for example, replacement of links with links of different flexibilities, can be done in minutes.

We have used this test-bed for experiments in hybrid force-position control of individual robots, and for multi-robot grasping; for the study of link compliance reported in this thesis and for the study of actuation compliance. This experimental setup has also been used for experimenting with human-robot interaction.

Figure A.1 shows the overall experimental setup. The two basic parts of the setup are the hardware and the software which runs the hardware. The hardware itself consists of two parts: the mechanical hardware comprising the fingers themselves, the tendon drive mechanism, the motors and the object being grasped and the electronic and computation hardware comprising the sensors, the amplifiers, the interface cards and the IBM compatible PC. The software is layered. The heart of the software is the scheduler servo, which runs the controller at the required

frequency. This is implemented in the Sparrow library [39]. The other layers of software are the libraries which implement primitives for fingers and for controllers. The user application is the top layer of the software which uses all others for actually running the experiment. In what follows we describe these in detail.

A.2 Hardware

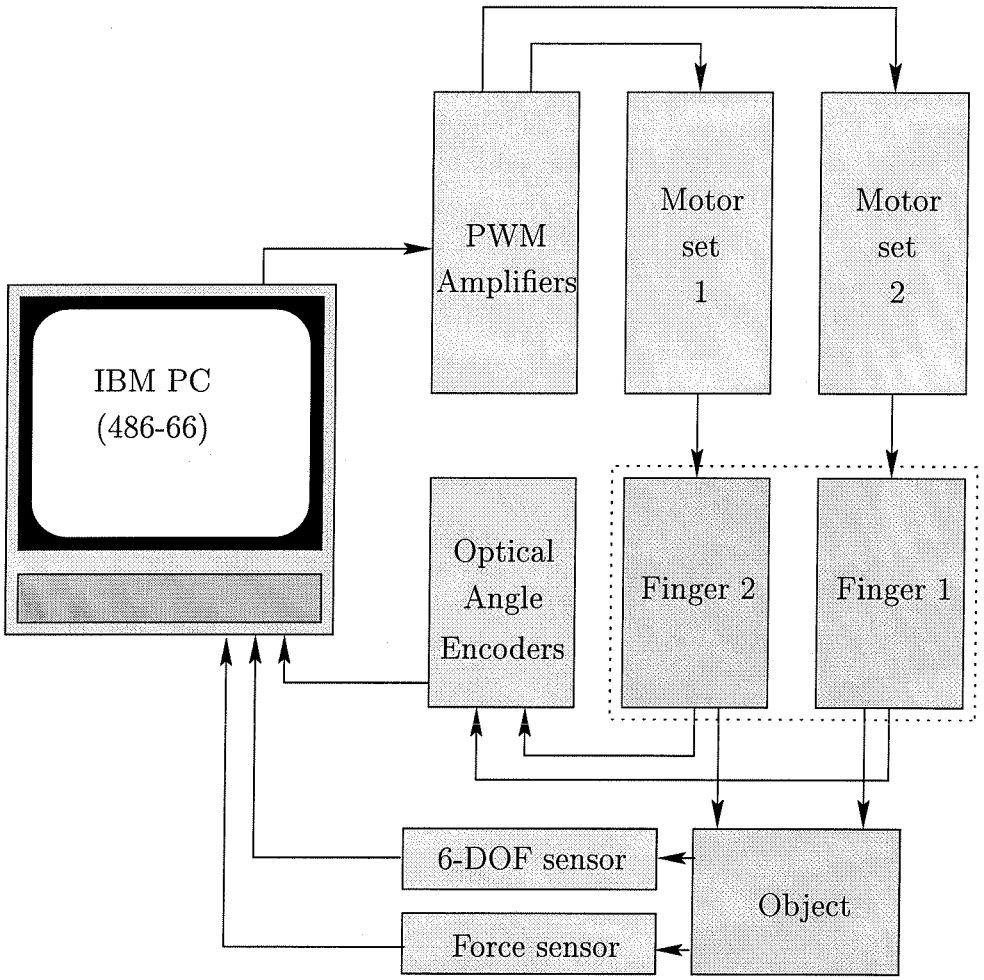


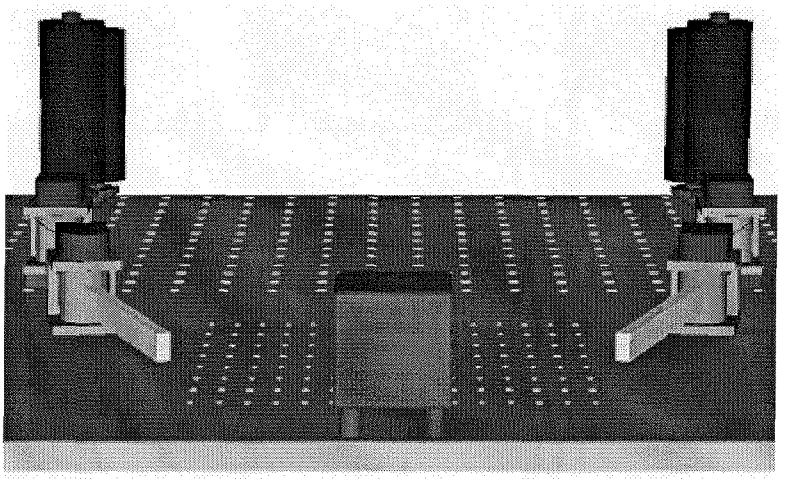
Figure A.2 Hardware setup.

The experimental setup used for our grasping experiments consists of two two-degree-of-freedom, revolute-jointed robot fingers. Figure A.2 is a block diagram of the hardware setup. The fingers are tendon driven with one motor driving each joint, a **1-n** configuration. The motors are driven using pulse-width-modulation (PWM) amplifiers. The fingers themselves have optical encoders at the joints for sensing the configuration of each finger. The object used for grasping is also instrumented

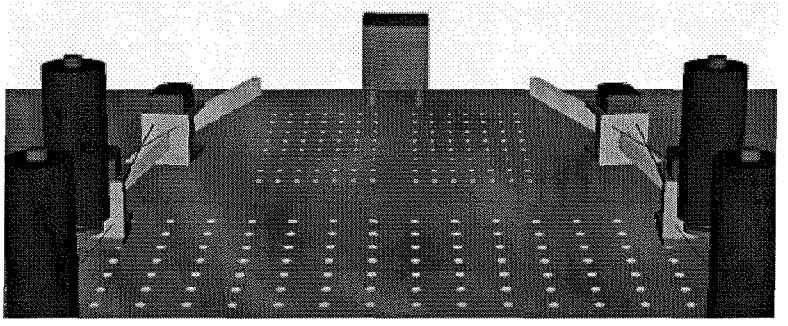
with a 6-DOF position and orientation sensor to get independent measurements of the object configuration. In addition there are force sensors on the grasped object to detect forces at the tips of the fingers as well as the internal force on the object. In what follows we describe various components of the setup in greater detail.

Base Plates: The testbed is erected on a set of two base plates which all together offer an area of size $26'' \times 24''$. The entire mechanical hardware for the setup is constructed on these base plates.

Metallic optical plate: One of the base plates is a standard, metallic optical plate with a grid of $1'' \times 1''$, $\frac{1}{4}$-20 tapped holes. It is $\frac{1}{2}$ inch thick. This plate has self leveling feet. The motors and finger base joints are mounted on this base plate (refer to Figure A.3), with optical bench type screwed down fixtures. Due to the use of this plate and the fixtures, the mechanical components can be relocated fairly easily to new locations.



(a) Front view.



(b) Back view.

Figure A.3 Mechanical hardware on base plates.

Non-metallic plate: The second base plate is a made of $\frac{1}{2}$ inch thick acrylic and has a pattern of calibration holes drilled into it with high precision. It mounts the transmitter of the Polhemus 3SPACE, InsideTRAK ™, 6-DOF position sensor. The Polhemus sensor is sensitive to metal and hence the requirement for a non-metallic mounting plate for this part of the setup. This plate is rigidly attached to the optical plate at its base and is leveled using leveling screws at its corners. The function of both these plates is to provide a stable, flat and level datum for the robotic setup. The non-metallic plate is also used to calibrate the position sensor.

The most important part of the robotic setup are the robots themselves. Each robot is a tendon actuated finger, with two revolute joints. In what follows we describe in detail the construction of the fingers.

Robotic fingers: Each finger is modular and is constructed by putting together joints and links in order. Though we use only two degree of freedom fingers, with two joints and two links for our experimentation, more complex robots can be built by using larger numbers of joints.

Links: For links any stock of rectangular cross-section, with the cross-sectional height $\frac{1}{2}''$ and thickness less than $\frac{2}{32}''$ can be used. The rigid link used during our experimentation was $\frac{2}{32}''(0.065'')$ thick and the most flexible link was $0.010''$ thick. The length of the link can vary.

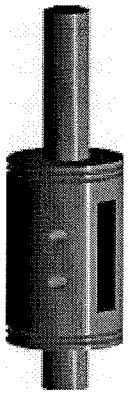
Joint Assembly: The joints are the crucial components in the make-up of the robotic fingers. They are not only the site where the macro internal movement of the fingers takes place, but are also the site of actuation. In addition, for tendon actuated fingers they are used for tendon routing. Each joint of the experimental setup is assembled from four different pieces.

Axle: The central axle is made of delrin. It has grooves at the top and the bottom to carry the tendons (Figure A.4a). The tendon routing is described later. The tendon actuating a joint makes a loop around the axle and through a slanted hole in the body of the axle, so that the two sides of the tendon do not rub against each other when being actuated. The base joints also guide the tendons for the next joint. For insertion of the link, the axle has a groove cut longitudinally into it. The link is fastened to the axle by means of a pair of set-screws. In the assembled joint the optical encoder is attached to the top shaft of the axle.

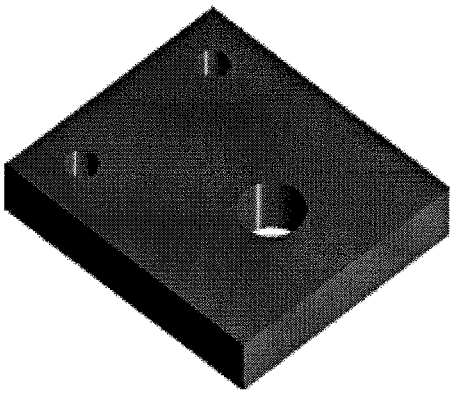
Base Plate: The axle rests on the base plate (refer Figure A.4b). The bottom shaft of the axle is inserted into the hole in the base plate. The base plate was originally designed without a bearing as we thought the delrin surfaces would provide sufficiently low friction. Later however bearings were put in to achieve the least possible amount of friction.

Back Plate: The back plate is the same height as central portion of the axle and serves to separate the base and the top plate. Additionally, the link from the previous joint is affixed to the back plate at a longitudinal groove and fastened by a pair of set-screws. It has apertures in it to allow the tendons to pass through. (Figure A.4c.)

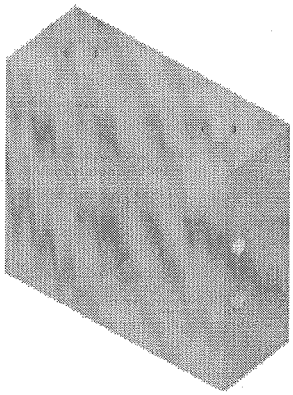
Top Plate: The top plate is similar to the base plate but in addition it provides the attachment surface for the optical encoders (Figure A.4d). The top shaft of the



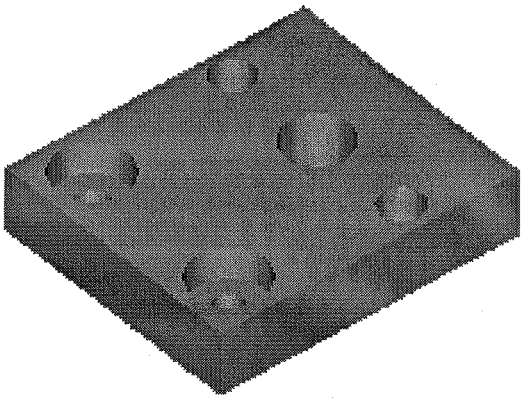
(a) Joint axle.



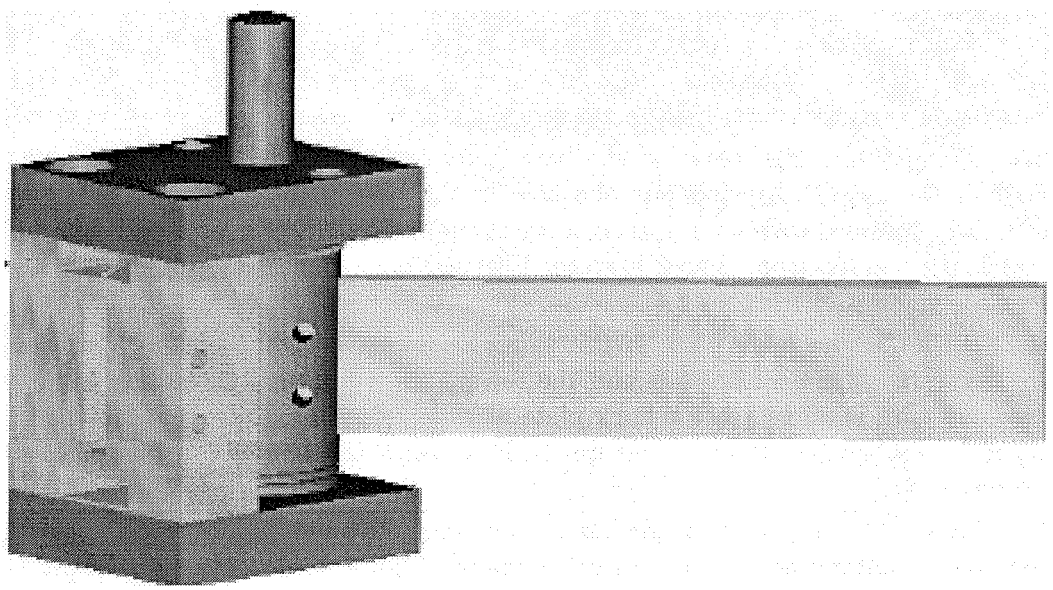
(b) Base plate.



(c) Back plate.



(d) Top plate.



(e) Assembled joint.

Figure A.4 Joint assembly.

axle protrudes through the top plate, and the optical encoder is attached between the top plate and the shaft.

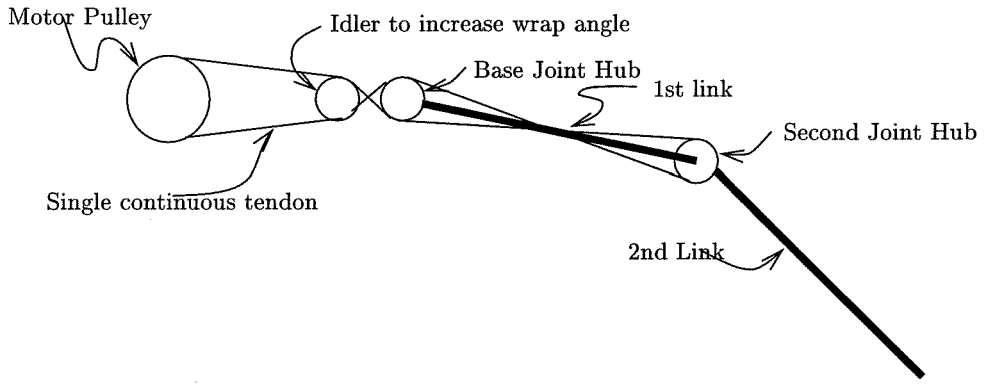
Figure A.4e shows the assembled joint with a link inserted into the axle. The optical encoders are attached at the top, over the protruding shaft in the figure.

Tendon driven manipulators are classified according to their actuation configuration. The more common configurations are **1-n** configuration, in which there is one motor actuating each joint, the **2-n** configuration which requires two motors for each joint and the **n+1** configuration which requires one motor more than the total number of actuated joints. It is important to realize that tendons can only transmit force in tension, which is why the various configurations are important. In the 1-n configuration, for each joint the tendons essentially make an endless loop between the joint and the motor pulley. For this configuration the length of the closed loop cannot change by very much during the motion of the finger. Tendon flexibility can make up for some change. The motor applies torque while moving both in the clockwise and anti-clockwise directions. In the 2-n configuration each motor for a joint applies torques in one direction only. When the joint rotates in a direction in which the motor is not applying torque, the motor is back-driven. This configuration also has no restriction on the change of length of the tendon path during motion, as the tendons do not form a closed loop. The disadvantage is the requirement of double the number of actuators, therefore requiring double the number of electronic hardware channels for control and amplification. The n+1 configuration uses one motor to apply torque in one direction to all the joints. The other motors apply torques on each joint in the direction opposite to that applied by the common motor. The effective torque is the difference in the torques applied by these motors. We used a 1-n configuration with the tendon routing described in what follows.

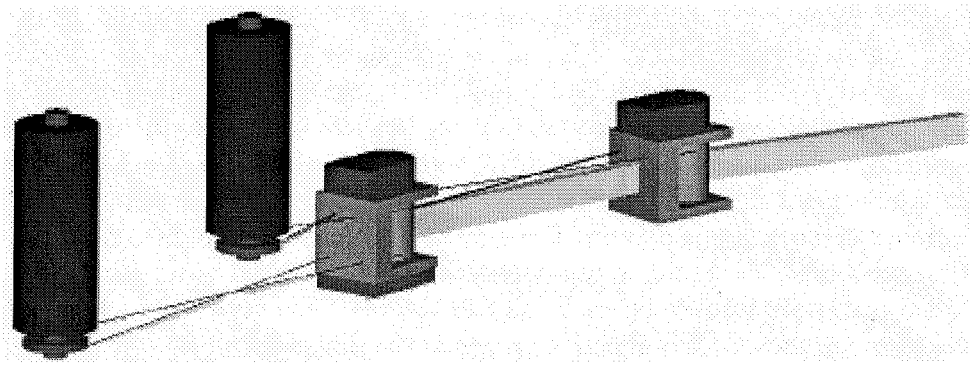
Tendon Routing: The tendon routing for the outer joint is done as shown in Figure A.5a. The tendons cross multiple times to increase the angle of wrap. They are staggered in height to avoid rubbing against each other at the crossing points. The routing for the base joint is similar (except for the last crossover as shown in Figure A.5b). The tendons are continuous loops from the motor pulley to the joint axle. At each joint axle, the tendons go over the grooves of the axle, through a slanted hole in the body of the axle and cross within the exit hole in the backplate of the joint as shown in Figure A.5c. Our tendon routing setup allows the tendons to be re-strung without requiring that fingers or motors be disassembled.

Tendon actuation introduces an additional transformation between the joint torques and the motor torques, parametrized by the configuration of the robot itself. The tendon routing we have used causes a coupling of the joint torques such that the motor torques required for each joint depends on the torque required by the other joints too. In Figure A.5a, the first three pulleys from the left are fixed relative to each other. However, the last pulley (the second joint) moves and can move far enough for the tendon to come off the base joint pulley on that side. The torque mapping changes when this happens, and must be taken into consideration during implementation of the experiment.

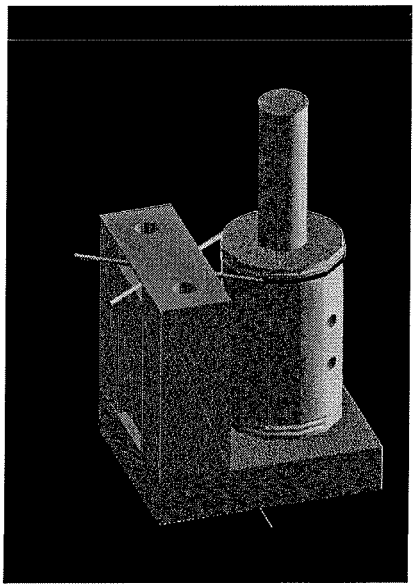
The remaining component of the fingers are the set of motors used to actuate



(a) Tendon routing: second joint.



(b) Tendon routing for finger.



(c) Tendon around axle.

Figure A.5 Tendon routing.

them. These must be matched to the specifics of the task required to be performed. Our setup allows changing motors quite easily, without having to disassemble the fingers themselves.

Motors: In grasping and hybrid force/position control tasks the usual scenario is one in which there are relatively small motions with relatively large forces. Thus the motors used to drive the fingers are usually stalled. However, they have large currents running through them to supply the torque necessary to maintain the end-effector force. This is potentially harmful to most DC motors and causes overheating and eventually burnout of the motor coils. The motors we used for our experiments were Maxon™ precision motors with precision, low backlash, single stage, planetary gearheads with a reduction ratio of 5.2. The motor/gearhead system delivered a stall torque of 2410 mNm and a load torque of 245 mNm. The voltage of operation was 24 volts and the maximum power was 90 watts. The motors were driven by Technology-80™ PWM amplifiers.

The other piece of mechanical hardware required for our experimentation was an instrumented object. For the grasping experiments it was required that the position and orientation of the grasped object be sensed independently of the configuration of the fingers, especially in case of the flexible fingers. Further, the internal force and the forces at the contact points were required to be known.

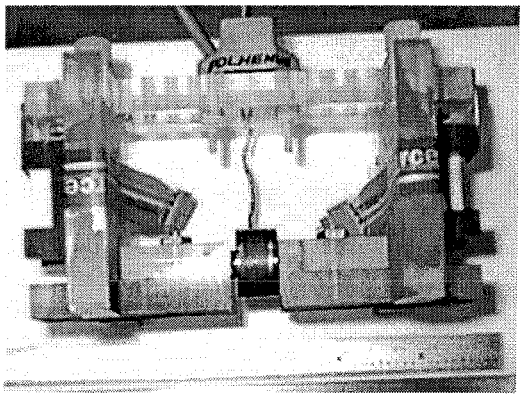


Figure A.6 Instrumented object.

Instrumented object: The instrumented object is made of lucite. It is constructed by joining together two vertical side plates with two horizontal struts. The top horizontal strut bears the receiver of the Polhemus 3PSACE, InsideTRAK™ six degree of freedom position and orientation sensor receiver. The lower strut was instrumented with a Sensotec™ one degree of freedom load cell for measurement of internal forces. Note that because of the presence of two struts the internal force detected by this sensor was not the total internal force, but a scaled version of it. To detect the forces of contact at the tips of the finger each contact point on the object was instrumented with a UniForce™ force sensor. The instrumented object is shown in Figure A.6.

In addition to the mechanical hardware described above, sensory hardware was required to sense the state of the experiment. We describe briefly the sensors used for experimentation.

Sensors and interface hardware: Three main types of sensors were used for experimentation. Each finger joint was instrumented with a HEDS™ optical encoder with 512 slits, a resolution of 0.1758°. Quadrature decoding for these sensors was done using HCTL-2016™ integrated circuits. The 3SPACE, InsideTRAK™ position and orientation sensing system, used for independently tracking the position and orientation of the object had a resolution of 0.0003 cms/cm in position measurement, 0.03° in angular measurement and a maximum update rate of 60 Hz. This sensor was interfaced to the computer with a manufacturer supplied DSP based interface card. The Sensotec™ single degree of freedom, analog load cell had a range of 0–1 kgf. The load cell had its own manufacturer supplied amplifier which conditioned and amplified the load cell voltage to ±5 volts. The UniForce™ force sensors had a range of 0–2 lbf. These sensors were based on force sensitive resistor technology and electronic hardware was required to be built for driving and amplifying the signals. Both types of force sensors were interfaced to the computer through a Keithley-Metrabyte DAS-1600™ I/O board with 8 differential (16 single ended) A/D inputs, 2 D/A outputs and 24 bits of parallel port. The A/D inputs had 12 bits of resolution and an overall bandwidth of 100kHz for all channels. A custom interface board was designed and built for interfacing the optical encoders and for generating the PWM signals for the motor amplifiers. This board was capable of reading and decoding 9 optical encoders and provided 8 channels of PWM output. The PWM carrier frequency was 10 kHz. The interface board was also capable of running multilevel scheduling loops for running nested or hierarchical controllers.

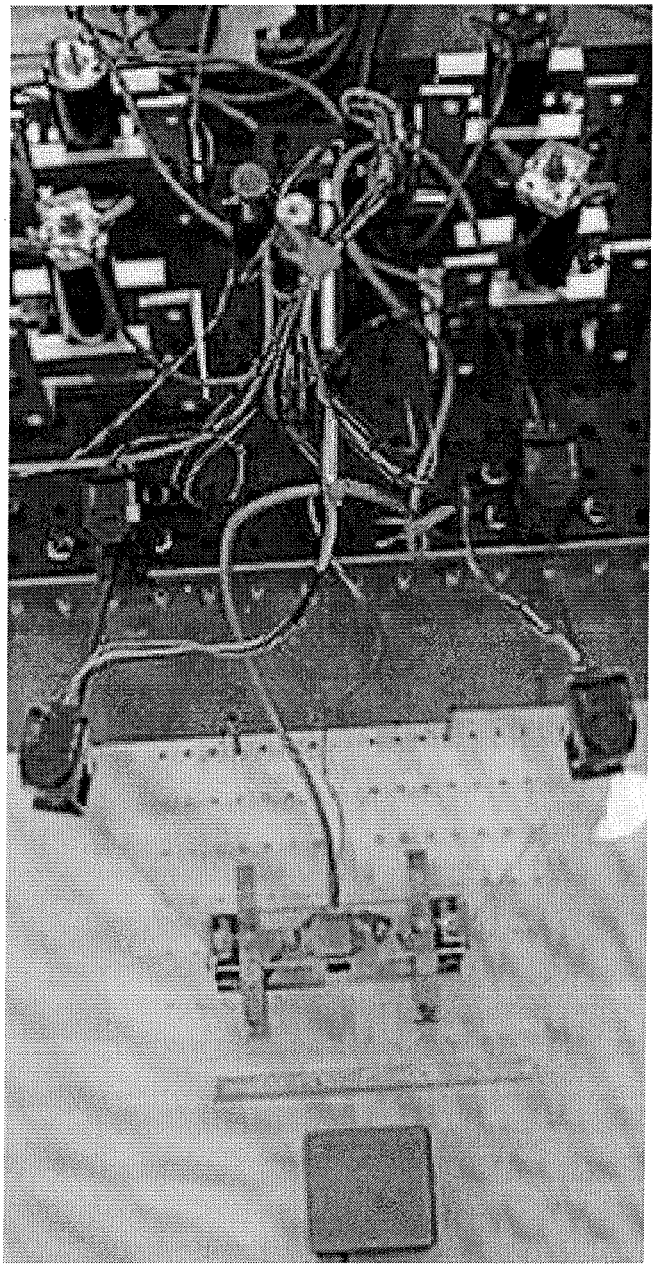
Photographs of the real setup are in Figure A.7.

In general tendon driven fingers are harder to control than direct drive fingers. However, there is a saving in weight of the manipulator which can lead to better performance. Additionally, tendon driven robots can be scaled down to small sizes more easily than direct drive robots because the actuators do not have to be scaled down.

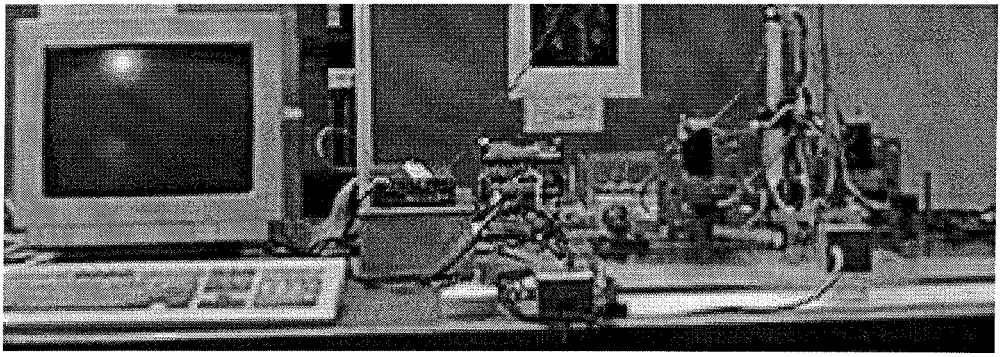
A.3 Software

We will only briefly discuss the software setup for experimentation. A detailed discussion is beyond the scope of this document and the reader is referred to [39] for details.

The software used for control of the robotic setup is based on the Sparrow real-time computation package [39] for IBM compatible PCs. The core of Sparrow is a scheduler which can be used to run the control loop at a very fixed frequency. In use the scheduler operates to execute the control function at the highest priority. Therefore, the control loop will interrupt every other "background" task which the computer is performing. The result is that the control loop is able to run exactly when called, without any latency (in principle). During a typical control



(a) Top view of fingers.



(b) View of experimental setup.

cycle sensors are read, control computations performed and the actuation desired output to actuators. Sparrow contains a large number of device drivers for reading various types of sensors and interface boards and for outputting actuator commands. In addition it implements a display manager which allows changing parameters interactively, and also serves as an medium for display of information about the experiment.

In addition to Sparrow, which is a very general software for doing real-time control, application specific software libraries were developed to facilitate programming. Primitives for robotic manipulation, like software definitions for individual finger dynamics and kinematics and two-finger grasps of individual objects were constructed for use in more complex programs. Primitives for implementation of robotic workspace controllers were also developed. These were used on top of the Sparrow library to write the overall code for running the experiments.

Bibliography

[1] J. Baumgarte. Stabilization of contraints and integrals of motion in dynamical systems. *Computer Methods in Applied Mechanics and Engineering*, pages 1–16, 1972.

[2] R. H. Cannon, Jr. and E. Schmitz. Initial experiments on the end-point control of a flexible one-link robot. *International Journal of Robotics Research*, 3(3):62–75, 1984.

[3] S. Cetinkunt and W. Yu. Accuracy of finite dimensional dynamic models of flexible manipulators for controller design. *Journal of Robotic Systems*, 9(3):327–350, 1992.

[4] A. B. A. Cole, J. E. Hauser, and S. S. Sastry. Kinematics and control of multifingered hands with rolling contact. *Proc. IEEE International Conference on Robotics and Automation*, pages 228–233, 1988.

[5] J. J. Craig and M. H. Raibert. A systematic method of hybrid position/force control of a manipulator. In *Proc. IEEE Computer Software and Applications Conference*, pages 446–451, 1979.

[6] J. H. Davis and R. M. Hirschorn. Tracking control of a flexible robot link. *IEEE Transactions on Automatic Control*, 33(3):238–248, 1988.

[7] J. Denavit and R. S. Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *Journal of Applied Mechanics*, pages 215–221, 1955.

[8] W. D. Fisher and M. S. Mujtaba. Hybrid position/force control: a correct formulation. *International Journal of Robotics Research*, 11(4):299–311, 1992.

[9] A. R. Fraser and R. W. Daniel. *Perturbation Techniques for Flexible Manipuilators*. Kluwer Academic Publishers, Boston, 1991.

[10] R. Frisch-Fay. *Flexible Bars*. Butterworths, London, 1962.

[11] Herbert Goldstein. *Classical Mechanics*. Addison-Wesley, Reading, Mass., second edition, 1981.

[12] K.W. Grace, J.E. Golgate, M.R. Glucksberg, and J.H. Chun. A six degree of freedom micromanipulator for ophthalmic surgery. In *Proc. IEEE International Conference on Robotics and Automation*, pages 630–635, 1993.

[13] N. Hogan. Impedance control: An approach to manipulation: Part I—theory. *ASME Journal of Dynamic Systems, Measurement and Control*, 107:1–7, March 1985.

[14] N. Hogan. Impedance control: An approach to manipulation: Part II—implementation. *ASME Journal of Dynamic Systems, Measurement and Control*, 107:8–16, March 1985.

[15] N. Hogan. Impedance control: An approach to manipulation: Part III—applications. *ASME Journal of Dynamic Systems, Measurement and Control*, 107:17–24, March 1985.

[16] F. C. Hoppensteadt. Singular perturbations on the infinite interval. *Transactions of the American Mathematical Society*, 123(2):521–535, June 1966.

[17] F. C. Hoppensteadt. On systems of ordinary differential equations with several parameters multiplying the derivatives. *Journal of Differential Equations*, 5:106–116, 1969.

[18] F. C. Hoppensteadt. Properties of solutions of ordinary differential equations with small parameters. *Communications on Pure and Applied Mathematics*, 24:807–840, 1971.

[19] K. Ishihara and T. Fukukawa. Intelligent microrobot DDS (Drug Delivery System) measured and controlled by ultrasonics. In *Proc. IEEE/RSJ International Workshop on Intelligent Robots and Systems; IROS '91*, 1991. Osaka, Japan.

[20] S. Jacobsen, J. Wood, K. Bigger, and E. Iverson. The Utah/MIT hand: Work in progress. *International Journal of Robotics Research*, 4(3):221–250, 1986.

[21] J. Kerr. *An Analysis of Multi-fingered Hands.* PhD thesis, Stanford University, Department of Mechanical Engineering, 1984.

[22] J. Kevorkian and J. D. Cole. *Perturbation Methods in Applied Mathematics.* Springer-Verlag, 1981.

[23] H. K. Khalil. *Nonlinear Systems.* Macmillan, 1992.

[24] P. V. Kokotovic. Applications of singular perturbation techniques to control problems. *SIAM Review, Society for Industrial and Applied Mathematics*, 26(4):501–550, Oct 1984.

[25] D. Kozel, A. J. Koivo, and S. S. Mahil. A general force/torque relationship and kinematic representation for flexible link manipulators. *Journal of Robotic Systems*, 8(4):531–556, 1991.

[26] D. J. Latornell and D. B. Chercas. Force and motion control of a single flexible manipulator link. *Robotics and Computer-Integrated Manufacturing*, 9(2):87–99, 1992.

[27] J. Lew and W. J. Book. Hybrid control of flexible manipulators with multiple contact. In *Proc. IEEE International Conference on Robotics and Automation*, pages 242–247, 1993.

[28] Z. Li, P. Hsu, and S. Sastry. On grasping and dynamic coordination of multifingered robot hands. Technical Report UCB/ERL M87/63, Electronicd Research Laboratory, University of California, Berkeley, Sep 1987.

[29] Z. Li, P. Hsu, and S. Sastry. On kinematics and control of multifingered hands. In *Proc. IEEE International Conference on Robotics and Automation*, pages 384–389, 1988.

[30] Z. Li, P. Hsu, and S. Sastry. Grasping and coordinated manipulation by a multifingered robot hand. *International Journal of Robotics Research*, 8(4):33–50, 1989.

[31] M. T. Mason. Compliance and force control for computer controlled manipulators. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-11(6):418–432, 1981.

[32] F. Matsuno, Y. Sakawa, and T. Asano. Quasi-static hybrid position/force control of a flexible manipulator. In *Proc. IEEE International Conference on Robotics and Automation*, pages 2838–2843, 1991.

[33] F. Matsuno and K. Yamamoto. Dynamic hybrid position/force control of a flexible manipulator. In *Proc. IEEE International Conference on Robotics and Automation*, pages 462–467, 1993.

[34] D. R. Meldrum, G. F. Franklin, and P. J. Wiktor. An inverse Jacobian solution for the control of multi-link flexible manipulators. In *Proc. American Control Conference*, pages 1814–1815, 1993.

[35] J. K. Mills. Hybrid control: A constrained motion perspective. *Journal of Robotic Systems*, 8(2):135–158, 1991.

[36] J. K. Mills. Stability and control aspects of flexible link robot manipulators during constrained motion tasks. *Journal of Robotic Systems*, 9(7):933–953, 1992.

[37] D. J. Montana. The kinematics of contact and grasp. *International Journal of Robotics Research*, 7(3):17–32, June 1988.

[38] D. J. Montana. The kinematics of contact with compliance. In *Proc. IEEE International Conference on Robotics and Automation*, pages 770–774, 1989.

[39] R. M. Murray. *Sparrow - Real Time Software for Control*, 1993.

[40] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1993.

[41] T. Okada. Object-handling system for manual industry. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-9(2):79–89, 1979.

[42] J. C. Piedbœuf and S. Miller. Estimation of endpoint position and orientation of a flexible link using strain gauges. In *Proc. Fourth IFAC Symposium on Robot Control*, pages 675–680, 1994.

[43] K. S. J. Pister. *Hinged Polysilicon Structures with Integrated CMOS Thin Film Transistors.* PhD thesis, Department of Electrical Engineering and Computer Science, University of California, Berkeley, 1992.

[44] K. S. J. Pister. Hinged polysilicon structures with integrated thin film transistors. In *Proc. IEEE Solid State Sensor and Actuator Workshop*, pages 136–139, 1992.

[45] M. Raibert and J. Craig. Hybrid position/force control of manipulators. *ASME Journal of Dynamic Systems, Measurement and Control*, 102:126–133, June 1981.

[46] R. M. Rosenberg. *Analytical Dynamics of Discrete Systems.* Plenum Press, New York, 1977.

[47] J. K. Salisbury. *Kinematic and Force Analysis of Articulated Hands.* PhD thesis, Stanford University, Department of Mechanical Engineering, 1982.

[48] J. K. Salisbury. Active stiffness control of a manipulator in cartesian coordinates. In *Proc. IEEE Control and Decision Conference*, pages 95–100, 1990.

[49] B. Siciliano and W. J. Book. A singular perturbation approach to control of fightweight flexible manipulators. *International Journal of Robotics Research*, 7(4):79–90, 1988.

[50] B. Siciliano, W. J. Book, and G. de Maria. An integral manifold approach to control of a one link flexible arm. In *Proc. 25th IEEE Conference on Decision and Control*, pages 1131–1134, 1986.

[51] V. A. Sobolev. Integral manifolds and decomposition of singularly perturbed systems. *Systems and Control Letters*, 5:169–179, December 1984.

[52] M. W. Spong, K. Khorasani, and P. V. Kokotovic. An integral manifold approach to the feedback control of flexible joint robots. *IEEE Journal of Robotics and Automation*, RA-3(4):291–300, August 1987.

[53] A. Tikhonov. On the dependence of solutions of differential equations on a small parameter. *Mat. Sb.*, 22(193–204), 1948. (In Russian).

[54] A. Tikhonov. Systems of differential equations containing a small parameter multiplying the derivative. *Mat. Sb.*, NS(31), 73:575–586, 1952. (In Russian).

[55] S. P. Timoshenko. *History of the Strength of Materials.* McGraw-Hill Book Company, Inc., 1953.

[56] P. B. Usoro, R. Nadira, and S. S. Mahil. A finite element/lagrange approach to modelling lightweight flexible manipulators. *ASME Journal of Dynamic Systems, Measurement and Control*, 108:198–205, Sep 1986.

[57] S. T. Venkataraman and T. E. Djaferis. Multivariable feedback control of the JPL/Stanford hand. In *Proc. IEEE International Conference on Robotics and Automation*, pages 77–82, 1987.

[58] M. Vidyasagar. *Nonlinear Systems Analysis*. Prentice-Hall, 1978.

[59] T. Yoshikawa and K. Hosoda. Modeling of flexible manipulators using virtual rigid links and passive joints. *International Journal of Robotics Research*, 15(3):290–299, June 1996.

[60] A. S. Zaki and W. H. El Maraghy. Modelling and control of a two-link flexible manipulator. *Canadian Society of Mechanical Engineers*, 16(3/4):311–328, 1992.

[61] H. Zhang and R. P. Paul. Hybrid control of robot manipulators. In *Proc. IEEE International Conference on Robotics and Automation*, pages 602–607, 1985.